

BAB IV

HASIL DAN PEMBAHASAN

Bab ini menguraikan hasil dan pembahasan dari implementasi serta uji coba dari sistem *automatic question generation* yang dibuat. Pada bab ini terdiri dari subbab pengumpulan data, perancangan model, pengembangan sistem, desain eksperimen, hasil eksperimen, dan analisis hasil.

4.1 Pengumpulan Data

Pada subbab ini menjelaskan bagaimana pengumpulan data dilakukan, data ini akan digunakan dalam penelitian ini. Data yang dikumpulkan berasal dari dua sumber yaitu *ebook* dan *website* IELTS.

Seperti yang terlihat pada Tabel 4.1, bahwa data soal yang terkumpul berasal dari *ebook* totalnya ada 49 dengan jumlah soal 924, sedangkan data soal yang terkumpul dengan sumber total *website* 4, ada 86 data soal yang terkumpul, jadi total data soal keseluruhan yang terkumpul berjumlah 1010. Data soal yang digunakan adalah contoh soal serta latihan soal yang ada pada buku ataupun *website* tersebut, dan soal yang diambil hanya soal dengan jenis *short answer question* pada *reading comprehension* IELTS.

Tabel 4. 1. Soal IELTS yang dikumpulkan dari *ebook* dan *website*.

Ebook		
No.	Sumber	Banyak Pertanyaan
1	<i>Cambridge Practice Tests for IELTS 2</i> (Cambridge University, 1996)	12
2	<i>Cambridge Practice Tests for IELTS 1</i> (Jakeman and McDowell, 1997)	5
3	<i>Cambridge IELTS 3 With Answer Edition</i> (Cambridge University, 2003)	10
4	<i>IELTS Reading Tests</i> (McCarter and Ash, 2001)	9
5	<i>Collins Reading for IELTS</i> (Geyte, 2011)	45
6	<i>Insight into IELTS Update Edition</i> (Jakeman and McDowell, 2001)	6
7	<i>Barron's IELTS</i> (Lougheed, 2006)	3

Ebook		
No.	Sumber	Banyak Pertanyaan
8	<i>Headway Academic Skills: Level 1 Student Book</i> (Harrison et al, 2013)	24
9	<i>Master IELTS 6: IELTS Precise Reading</i> (Patrick, 2001)	25
10	<i>Focusing on IELTS: Reading and Writing Skills</i> (Lindeck et al, 2000)	25
11	<i>Preparation and Practice: Reading and Writing General Training Module</i> (Pejovic et al, 1999)	33
12	<i>The Official Cambridge Guide to IELTS</i> (Cullen et al, 2014)	28
13	<i>IELTS Target 6.5: Course Book and Workbook</i> (Gough, 2013)	20
14	<i>IELTS Target 5.0 Leading to IELTS Academic: Course Book</i> (Gough, 2011)	33
15	<i>IELTS Reading Formula</i> (Memarzadeh, 2016)	80
16	<i>Lessons for IELTS Reading</i> (Group IELTS Research Institute, 2013)	36
17	<i>Essential Reading for IELTS</i> (Min and Gordon, 2014)	19
18	<i>IELTS Trainer Six Practice Tests</i> (Hashemi and Thomas, 2011)	4
19	<i>IELTS Masterclass: Student's Book</i> (Haines and May, 2011)	15
20	<i>IELTS Practice Tests</i> (Rehman, 2015)	10
21	<i>Mission IELTS Workbook 1</i> (Obee and Spratt, 2011)	15
22	<i>101 Helpful Hints for IELTS Academic Module</i> (Adams and Peck, 2000)	12
23	<i>Achieve IELTS Practice Test Book</i> (Bazin and Boyd, 2008)	7
24	<i>Achieve IELTS 2: Student's Book</i> (Harrison et al, 2006)	35
25	<i>Achieve IELTS 2: Workbook</i> (Harrison et al, 2006)	12
26	<i>A Book for IELTS 2nd Edition</i> (McCarter et al, 2003)	15
27	<i>Succeed in IELTS 9 Practice Tests</i> (Betsis and Mamas, 2014)	7
28	<i>IELTS The Complete Guide to Academic Reading</i> (Biggerton, 2014)	33

Yahya Firdaus, 2020

**AUTOMATIC GENERATE QUESTION UNTUK SHORT-ANSWER-QUESTION DI READING
COMPREHENSION IELTS MENGGUNAKAN METODE NATURAL LANGUAGE PROCESSING DAN
ALGORITMA K-NEAREST NEIGHBOR**

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

Ebook		
No.	Sumber	Banyak Pertanyaan
29	<i>The New Prepare for IELTS Academic Modules</i> (Cameron and Todd, 2007)	4
30	<i>Action Plan for IELTS</i> (Jakeman and McDowell, 2006)	6
31	<i>Bridge to IELTS: Student's Book</i> (Harrison and Hutchison, 2013)	13
32	<i>Bridge to IELTS: Workbook</i> (Rogers, 2013)	24
33	<i>Prepare For IELTS General Training Practice Tests</i> (Cameron and Todd, 2006)	19
34	<i>Get Ready for IELTS: Student's Book</i> (Aish et al, 2016)	27
35	<i>Get Ready for IELTS Reading</i> (Geyte, 2012)	26
36	<i>Focus On Academic Skills for IELTS</i> (Terry and Wilson, 2004)	6
37	<i>How to Prepare for IELTS New Edition</i> (Witt, 2001)	7
38	<i>High Impact IELTS Academic Module</i> (Bourne, 2004)	105
39	<i>IELTS Express Upper Intermediate Coursebook Second Edition</i> (Hallows et al, 2005)	6
40	<i>Essential Tests for IELTS</i> (Griffiths, 2014)	13
41	<i>IELTS for Academic Purposes: 6 Practice Tests</i> (Mann and Taylore, 2009)	11
42	<i>IELTS for Academic Purposes: A Short Intensive Course</i> (Mann and Taylore, 2009)	5
43	<i>IELTS Express Upper Intermediate Workbook</i> (Humphreys and Cullen, 2006)	5
44	<i>IELTS Express Upper Intermediate Coursebook</i> (Hallows et al, 2006)	7
45	<i>IELTS Advantage Reading Skills</i> (Taylor and Wright, 2012)	16
46	<i>Mission IELTS Academic 1: Student's Book</i> (Obee and Spratt, 2010)	16
47	<i>Cambridge IELTS 6</i> (Cambridge University, 2007)	4
48	<i>IELTS Reading 100% 9 Points</i> (Cambrigton, 2014)	10
49	<i>Exam Essentials IELTS Practice Test 1 with Key</i> (Harrison and Whitehead, 2014)	16
TOTAL		924

Yahya Firdaus, 2020

**AUTOMATIC GENERATE QUESTION UNTUK SHORT-ANSWER-QUESTION DI READING
COMPREHENSION IELTS MENGGUNAKAN METODE NATURAL LANGUAGE PROCESSING DAN
ALGORITMA K-NEAREST NEIGHBOR**

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

Website		
No.	Sumber	Banyak Pertanyaan
1	https://www.ieltshelpnow.com	12
2	https://www.ieltsbuddy.com/ielts-short-answer-questions.html	6
3	https://www.ieltsmaterial.com	58
4	https://www.engexam.info	10
TOTAL		86

Soal-soal yang telah dikumpulkan kemudian diproses untuk mendapatkan banyak kata dan POS *tag* pada setiap kalimat soal. Data banyak kata pada setiap kalimat ini digunakan untuk mencocokkan dengan data uji, agar perbandingan yang dilakukan sesuai setiap jumlah tag. Data POS tag digunakan untuk menentukan jarak kesesuaian dengan data uji.

Tabel 4. 2. Representasi data soal

No	Pertanyaan	POS tag	Banyak Kata
1			
...			
1010			

Pada Tabel 4.2, merupakan representasi dari data soal yang akan digunakan dalam penelitian ini. Setiap soal ini dimasukan ke sistem dalam bentuk input file, sehingga dataset soal ini dapat diperbaharui agar hasilnya dapat lebih layak untuk menjadi soal tes. Lebih jelasnya mengenai Tabel 4.2, adalah sebagai berikut:

1. No. Kolom ini digunakan untuk menjelaskan nomor dari data soal. Nomor ini dipresentasikan dari 1 hingga 1010 atau sebanyak data soal.
2. Pertanyaan. Pada kolom ini merupakan soal-soal utuh yang diambil dari contoh soal maupun latihan soal yang ada dalam ebook, yang kemudian dimasukan dalam dataset. Soal yang dimasukan dalam dataset tidak dirubah, dari kata sampai simbol-simbol tidak ada yang dihapus ataupun dirubah.
3. POS tag. Soal-soal dari dataset di identifikasi POS tag oleh sistem, sehingga didapat POS tag dari setiap kata pada kalimat yang dipisahkan oleh spasi.

POS *tag* ini digunakan untuk membandingkan dengan data uji, yang digunakan untuk mengukur kelayakan data uji.

4. Banyak Kata. Kolom ini dipakai untuk menentukan jumlah kata pada kalimat, karena setiap banyak kata pada setiap kalimat soal berbeda, dan untuk membandingkan dengan data uji diperlukan banyak kata yang sama. Oleh karena itu banyak kata dalam kalimat diperlukan. Sebelum proses POS *tag*, data-data soal dibersihkan terlebih dahulu dari simbol-simbol, sehingga banyak POS *tag* sesuai dengan banyak kata.

Tabel 4. 3. Contoh sampel tiga baris koleksi IELTS

No	Pertanyaan	POS <i>tag</i>	Banyak Kata
1	Who found the shoe?	WP VBD DT NN	4
2	When was puffery first used?	WRB VBD NN RB VBN	5
3	What's his new phone number?	VBZ PRP\$ JJ NN NN	5

Tabel 4.3, merupakan beberapa contoh dari data soal yang diambil dari ebook dan website, jumlah total data soal yang dikumpulkan sebanyak 1010. Data soal yang dikumpulkan merupakan soal dengan jenis *short answer question* di *reading comprehension* IELTS. Data yang ditampilkan pada Tabel 4.3, merupakan contoh dari sebagian data yang akan digunakan oleh penulis dalam tahapan proses selanjutnya.

4.2 Perancangan Model

Tahapan dalam perancangan model sistem *automatic question generation* ini tentunya melalui tahapan-tahapan sampai menghasilkan soal dengan jenis soal *short answer question*. Tahapan yang ada pada pengembangan sistem ini berbeda dengan enam tahapan pembuatan IELTS secara konvensional, yaitu antara lain *commissioning*, *pre-editing*, *editing*, *pretesting*, *standard fixing*, dan *test construction and grading*. Tahapan pengembangan sistem ini diluar dari enam tahapan tersebut dengan pendekatan yang berbeda, sehingga tidak ada irisan untuk enam tahapan pembuatan soal IELTS secara konvensional dengan tahapan yang ada pada pengembangan sistem ini.

Yahya Firdaus, 2020

AUTOMATIC GENERATE QUESTION UNTUK SHORT-ANSWER-QUESTION DI READING COMPREHENSION IELTS MENGGUNAKAN METODE NATURAL LANGUAGE PROCESSING DAN ALGORITMA K-NEAREST NEIGHBOR

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

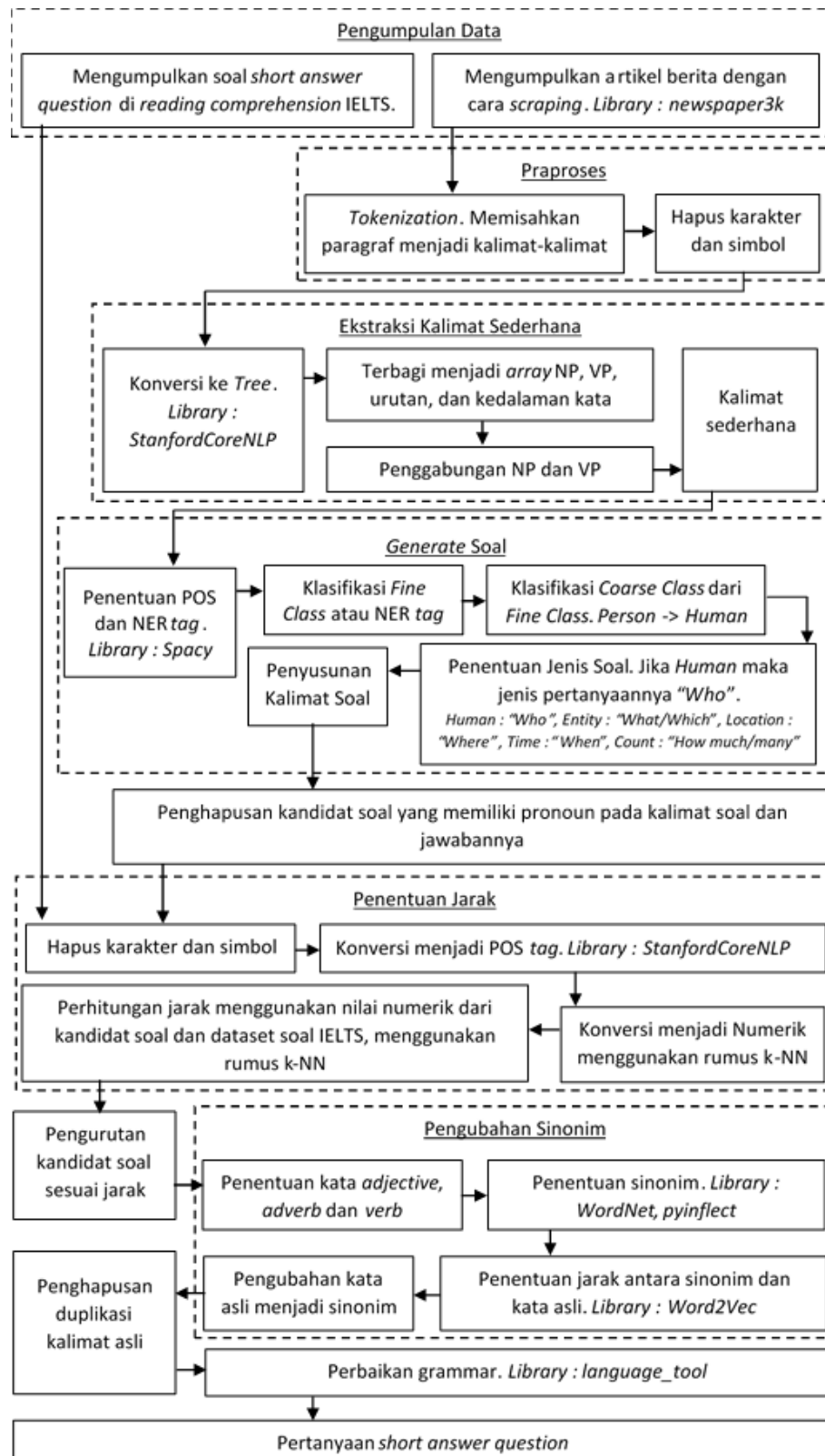
Beberapa tahapan yang ada pada pengembangan sistem ini yaitu praproses paragraf dari artikel dan pemisahan per kalimat, pembuatan *tree structure* dari kalimat-kalimat yang sudah di praproses, penyederhanaan kalimat dari kalimat kompleks, tahapan penentuan *course class* atau aturan dari kalimat tersebut seperti “*human verb human*”, pengklasifikasian jenis soal dari aturan kalimat, tahapan *generate* soal dari kalimat menggunakan jenis-jenis soal yang sudah didapat dari tahapan sebelumnya, selanjutnya adalah tahapan rangking dari kalimat, tahapan penentuan jarak dari data uji atau soal yang di *generate* dengan data latih, dan penentuan kelayakan.

Tahapan-tahapan tersebut diperlukan untuk menghasilkan soal *short answer question* yang layak untuk reading *comprehension* IELTS. Pada Gambar 4.1, dijelaskan dengan detail per tahapan dalam perancangan pembuatan sistem *automatic question generation* yang penulis lakukan. Pada Gambar 4.1, dapat diketahui bahwa untuk *library* yang digunakan yaitu hanya untuk melakukan *scraping*, melakukan konversi dari kalimat menjadi *tree*, mendapatkan POS dan NER *tag*, penentuan sinonim, dan juga untuk pengecekan *grammar*. Penjelasan singkat pada Gambar 4.1, dari setiap proses yang dilakukan dapat dilihat pada Tabel 4.5, sedangkan untuk penjelasan mengenai *library* yang digunakan serta kegunaannya dapat dilihat pada Tabel 4.4. Dapat diketahui dari Tabel 4.4, bahwa *library* yang digunakan dalam penelitian ada tujuh *library*.

Tabel 4. 4. Penjelasan *library* yang digunakan dalam sistem.

No	<i>Library</i>	Kegunaan
1	newspaper3k	Digunakan untuk melakukan <i>scraping</i> atau mengambil isi artikel dari <i>link</i> artikel yang dimasukkan.
2	StanfordCoreNLP	Digunakan untuk melakukan konversi dari kalimat menjadi <i>tree structur</i> .
3	Spacy	Digunakan untuk mendapatkan NER <i>tag</i> dan POS <i>tag</i> dari kalimat.
4	WordNet	Digunakan untuk mendapatkan sinonim dari suatu kata

No	Library	Kegunaan
5	Pyinflect	Digunakan untuk mengubah suatu kalimat menjadi sesuai dengan POS <i>tag</i> nya. Contohnya mengubah kata <i>verb</i> sesuai dengan jenisnya.
6	Word2Vec	Digunakan untuk mendapatkan nilai jarak antara kata sinonim dengan kata asli.
7	language_tool	Digunakan untuk melakukan pengecekan <i>grammar</i> dan memperbaikinya.



Gambar 4.1 Alur Model Sistem Penghasil Soal Short Answer Question

Yahya Firdaus, 2020

AUTOMATIC GENERATE QUESTION UNTUK SHORT-ANSWER-QUESTION DI READING COMPREHENSION IELTS MENGGUNAKAN METODE NATURAL LANGUAGE PROCESSING DAN ALGORITMA K-NEAREST NEIGHBOR

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

Tabel 4. 5. Penjelasan singkat dari setiap proses yang dilakukan.

No.	Proses	Kegunaan
1	<i>Scrpping</i>	Proses ini bertujuan untuk mendapatkan isi artikel dari link url <i>website</i> yang dimasukan
2	<i>Tokenization</i>	Bertujuan untuk memisahkan paragraf menjadi kalimat-kalimat
3	Konversi menjadi <i>tree</i>	Mengubah kalimat menjadi <i>tree structure</i>
4	Ekstraksi kalimat sederhana	Proses ini bertujuan untuk mendapatkan kalimat sederhana dari kalimat-kalimat kompleks
5	Generate soal	Proses ini bertujuan untuk menghasilkan kalimat soal beserta jawabannya dari kalimat yang dimasukan
6	<i>Clear pronoun</i>	Pada proses ini kalimat dan jawaban soal dipilah dari <i>pronoun</i> .
7	Konversi menjadi <i>tag</i>	Bertujuan untuk mendapatkan <i>tag</i> dari setiap kata dalam kalimat soal
8	Konversi <i>tag</i> menjadi numerik	Digunakan untuk mendapatkan nilai numerik dari setiap <i>tag</i> dalam kalimat
9	Menghitung jarak	Menghitung jarak antara data uji dengan data latih menggunakan algoritma K-NN
10	<i>Sorting</i>	Melakukan <i>sorting</i> sesuai dengan kedekatan atau dilihat dari jarak terkecil
11	Mengubah kalimat	Proses ini mengubah kalimat menjadi kalimat yang berbeda dengan mengganti <i>verb</i> , <i>adverb</i> , <i>adjective</i> dengan sinonimnya
12	Cek <i>grammar</i>	Bertujuan untuk melakukan cek <i>grammar</i> dari kalimat soal yang dihasilkan
13	<i>Clear duplicate</i>	Proses ini bertujuan untuk menghapus <i>duplicate</i> dari kalimat yang jadi kalimat soal

4.2.1 Pengumpulan *ebook* IELTS

Tahapan awal dari model *automatic question generation* ini adalah pengumpulan soal-soal *reading comprehension* IELTS dengan jenis *short answer question*, soal-soal yang dikumpulkan ini yang nantinya akan digunakan sebagai data latih untuk menguji kelayakan dari kandidat soal. Hasil dari tahapan ini berupa kumpulan contoh soal yang dapat dijadikan pembanding kelayakan dengan data uji

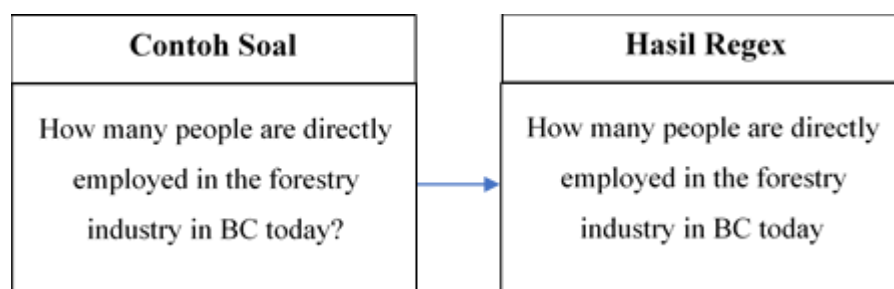
atau kandidat soal. Lebih jelasnya dari tahapan ini ada pada subbab 4.1, mengenai bagaimana pengumpulan soal dan pemrosesan dilakukan.

4.2.2 Praproses Data Soal IELTS

Setelah soal-soal yang dikumpulkan telah cukup untuk menjadi data latih, tahap selanjutnya adalah melakukan praproses pada data soal. Karena soal yang telah dikumpulkan pada tahapan sebelumnya masih dalam bentuk soal utuh dengan simbol-simbol, maka perlu dilakukannya praproses agar data soal dapat digunakan untuk data latih.

Tujuan dari praproses ini adalah menghilangkan simbol-simbol yang tidak dapat dibaca oleh *Part of Speech* (POS). Menghilangkan simbol-simbol ini penulis menggunakan *regular expression* atau biasa dikenal dengan sebutan regex. Regex ini sendiri termasuk kedalam *Natural Language Programming* (NLP), yaitu pengolahan teks. Penjelasan mengenai regex telah dijelaskan pada subbab sebelumnya, dan akan diimplementasikan pada tahapan ini.

Karakter yang dihilangkan berupa karakter atau simbol yang tidak dapat dibaca oleh POS *tag*, dilakukan pada semua dataset soal, yaitu berjumlah 1010 soal. Potongan kode tersebut juga dilakukan pada data soal uji yang menjadi kandidat soal. Contoh dari hasil praproses dapat terlihat pada Gambar 4.2. Dapat diperhatikan bahwa karakter seperti koma (,) atau tanda tanya (?) sudah dihapus. Setelah dataset soal latih telah bersih dari karakter dan simbol, maka dapat lanjut kelangkah selanjutnya.



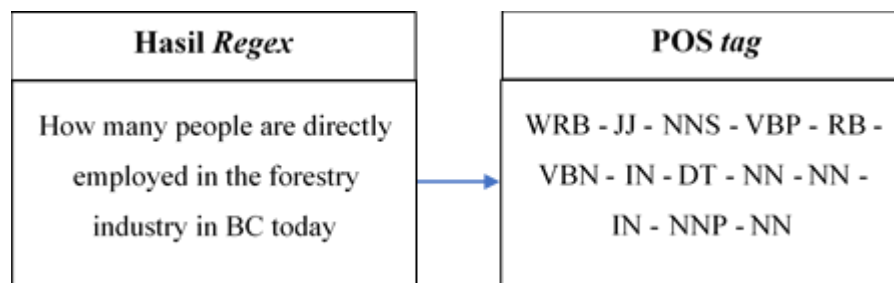
Gambar 4. 2 Contoh Proses Menghilangkan Karakter dan Simbol.

4.2.3 *Part of Speech Tagging* Soal IELTS.

Selanjutnya setelah tahapan praproses adalah melakukan *tagging* pada tiap kata dalam kalimat yang ada pada dataset soal. Penjelasan mengenai POS *tag* telah

dijelaskan sebelumnya. *POS tag* ini dilakukan untuk mengetahui jenis tiap kata yang sesuai dengan fungsi tiap kata pada kalimat. Selain melakukan *tagging*, juga dihitung banyak kata dalam kalimat melalui perhitungan banyaknya tag yang dihasilkan dalam satu kalimat soal, perhitungan banyak kata dalam satu kalimat ini berguna untuk tahapan selanjutnya. Contoh hasil dari tahapan ini dapat dilihat pada Gambar 4.4, hasil ini sudah sesuai dengan yang dijelaskan pada subbab sebelumnya.

Dapat dilihat pada Gambar 4.4, kalimat soal hasil regex yang sudah tidak memiliki karakter maupun simbol dikonversi menjadi *POS tag* serta urutan dan banyak tag hasil dari konversi memiliki jumlah yang sama dengan klaimat soal sebelum dikonversi. Contohnya seperti kata “*When*” yang memiliki tag “*WRB*” yaitu *wh-adverb*, dan kata “*puffery*” yang dikonversi *POS tag* menjadi “*NN*” yaitu *noun singular or mass*. Juga dapat dihitung jumlah tag atau kata nya adalah 5, maka banyak kata untuk kalimat soal ini adalah 5 dengan *POS tag* seperti yang terlihat pada Gambar 4.4. *Library* yang digunakan pada proses ini adalah *StanfordCoreNLP*, yang mana digunakan untuk mengkonversi kalimat menjadi *tree*.



Gambar 4.3 Contoh *POS tag* dari hasil *regex*.

4.2.4 Pengumpulan Artikel

Tahapan ini diperlukan untuk mengumpulkan artikel yang akan diekstrak menjadi beberapa soal yang layak digunakan sebagai jenis soal *short answer question* yang ada pada *reading comprehension* IELTS. Artikelnya dapat disalin dari artikel berita maupun jurnal ilmiah, dapat disesuaikan dengan jenis tes IELTS. Seperti tes IELTS *module General Training* yang materi dari artikel berupa kehidupan social, dan tes IELTS *module Academic* yang materi dari artikelnya berupa artikel ilmiah. Gambar 4.4, menampilkan contoh artikel yang diambil dari

Yahya Firdaus, 2020

AUTOMATIC GENERATE QUESTION UNTUK SHORT-ANSWER-QUESTION DI READING COMPREHENSION IELTS MENGGUNAKAN METODE NATURAL LANGUAGE PROCESSING DAN ALGORITMA K-NEAREST NEIGHBOR

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

website berita BBC yaitu mengenai kondisi corona yang terjadi di India. *Library* yang digunakana pada tahapan ini adalah newspaper3k yang digunakan untuk melakukan pengambilan isi artikel dari *link* yang dimasukkan oleh pengguna.

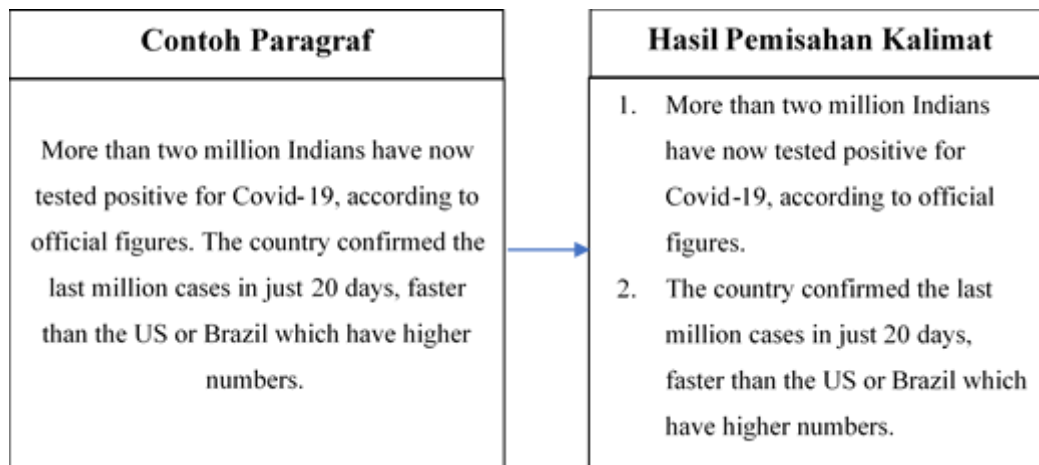
More than two million Indians have now tested positive for Covid-19, according to official figures. The country confirmed the last million cases in just 20 days, faster than the US or Brazil which have higher numbers. Testing has been expanded considerably in India in recent weeks but the situation varies across states. Spurred by a low death rate, the nation continues to reopen even as new hotspots drive the surge in cases. But some states have imposed restrictions. The recent measures include local, intermittent lockdowns, sometimes limiting activity in specific cities or districts. India is now the third country to cross the two million mark. It reported 62,170 cases in the past 24 hours, taking its total tally up to 2,025,409. The country has reported around 40,700 deaths so far. While that is the world's fifth-biggest total, experts say it is not very high given the country's population of 1.3 billion.

Gambar 4. 4 Contoh artikel berita yang diambil dari *website*.

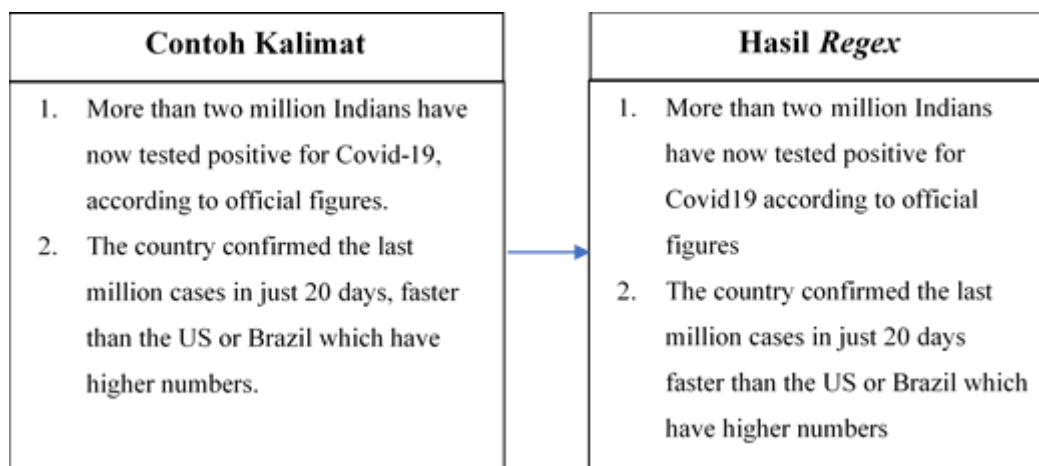
4.2.5 Pemisahan Kalimat dan Praproses dari Paragraf

Setelah didapatkan artikel yang akan dikonversi menjadi beberapa soal, langkah selanjutnya adalah melakukan pemisahan kalimat. Pemisahan ini dilakukan dengan syarat yaitu awal dari kalimat haruslah diawali dengan huruf kapital dan diakhiri dengan titik, jika tidak memenuhi syarat maka kalimat tersebut tidak akan diproses untuk ke langkah selanjutnya, syarat pemisahan kalimat ini dijelaskan oleh Ali (2010) dalam penelitiannya. Gambar 4.5, menampilkan contoh hasil pemisahan kalimat dari paragraf.

Langkah selanjutnya setelah melakukan pemisahan kalimat adalah melakukan praproses pada setiap kalimat yang dihasilkan dari paragraf. Praproses ini dilakukan dengan tujuan menghilangkan karakter dan simbol-simbol, agar dapat diproses pada tahapan selanjutnya. Menghilangkan karakter dan simbol ini menggunakan fungsi *regex*. Contoh hasil dari praproses dapat dilihat pada Gambar 4.6.



Gambar 4.5 Contoh hasil pemisahan kalimat dari paragraf.



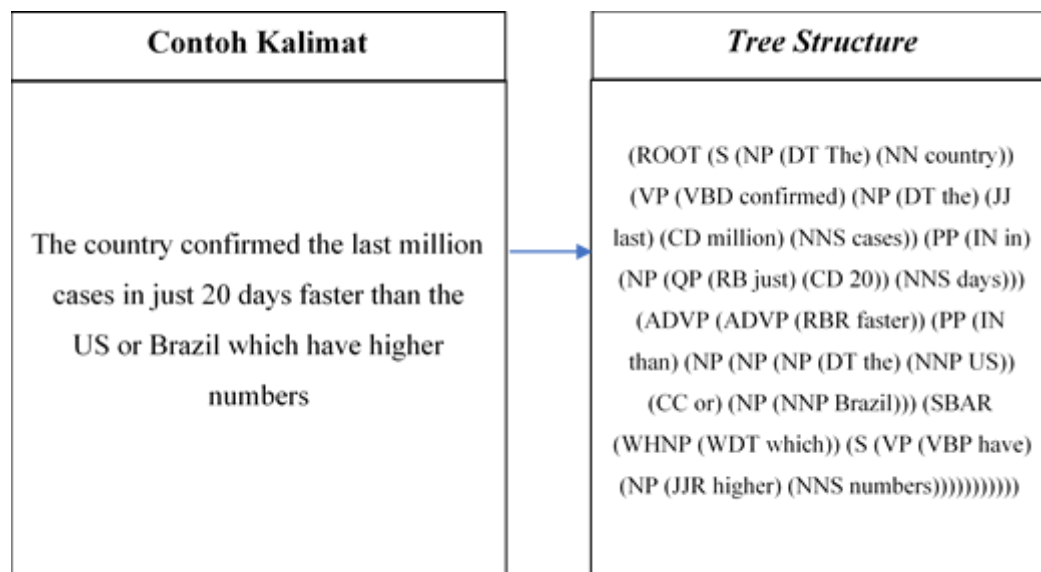
Gambar 4.6 Contoh hasil menghilangkan karakter dan simbol.

4.2.6 Konversi *Tree Structure*

Pada tahapan ini kalimat yang sudah dibersihkan dari karakter atau simbol menggunakan *regex* tahapan sebelumnya akan dikonversi menjadi *tree structure*, karena karakter ataupun simbol tidak bisa dikonversi ke POS tag, maka *regex* ini diperlukan. *Library* yang digunakan untuk melakukan konversi menjadi *tree structure* ini menggunakan StanfordCoreNLP melalui NLTK, dan menggunakan bahasa pemrograman python.

Hasil dari konversi menjadi *tree* ini akan dipakai dalam tahapan selanjutnya untuk melakukan ekstraksi kalimat sederhana dari kalimat kompleks. Contoh hasil dari konversi menjadi *tree structure* dapat dilihat pada Gambar 4.7. Sebelumnya

hasil dari konversi ini dalam bentuk data *tree*, oleh karena itu diperlukan proses konversi dari tipe data *tree* menjadi *string*.

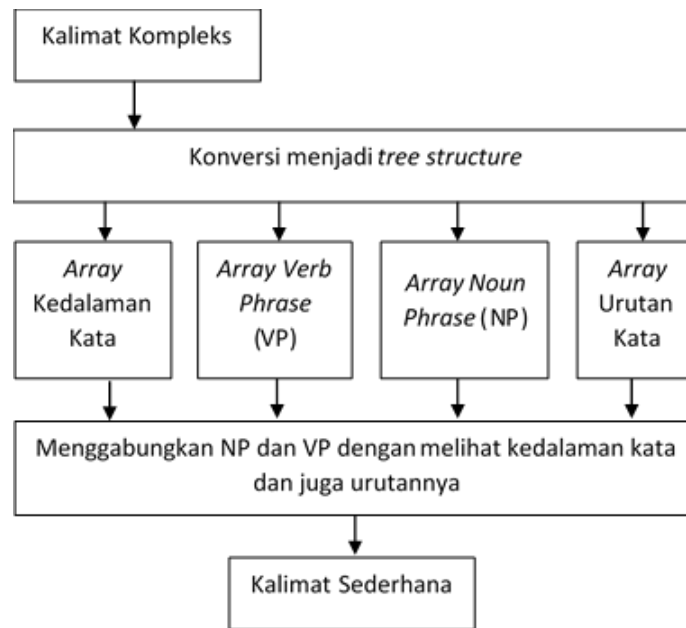


Gambar 4. 7 Contoh hasil konversi menjadi *tree structure*.

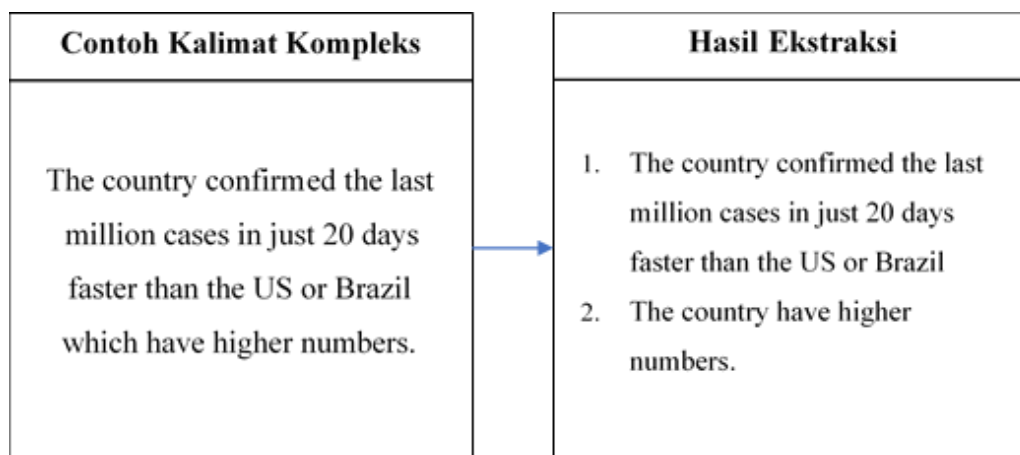
4.2.7 Pembuatan Sistem ESES

Sistem *Elementary Sentence Extracted System* (ESES) merupakan sistem yang berguna untuk melakukan ekstraksi kalimat elementer atau kalimat sederhana dari kalimat kompleks. Karena dalam pembuatan soal diperlukan kalimat yang sederhana, maka ESES akan sangat berguna dalam hal ekstraksi kalimat ini. Gambar 4.8, merupakan alur model penelitian yang dilakukan oleh Ali (2010) untuk membuat ESES, dari kalimat sederhana dibagi menjadi empat *array* yaitu diantaranya NP atau *noun phrase*, VP atau *verb phrase*, kedalaman kata didalam kalimat, dan juga urutan kata pada kalimat.

Yang kemudian setiap NP dan VP akan digabungkan dengan melihat kedalaman serta urutan setiap NP dan VP. Akan tetapi kalimat yang dianggap kalimat kompleks yaitu kalimat yang memiliki *subject* lebih dari satu. Sedangkan kalimat sederhana hanya memiliki satu *subject*, tidak kurang dan tidak lebih. Syarat ini sesuai dengan penelitian yang dilakukan oleh Kalady dkk (2010). Pada Gambar 4.9, memperlihatkan contoh hasil ekstraksi kalimat sederhana dari kalimat kompleks.



Gambar 4. 8 Model alur Elementary Sentence Extracted System (Ali, 2010).



Gambar 4. 9 Contoh hasil ekstraksi kalimat kompleks menjadi kalimat sederhana.

4.2.8 Penentuan *Fine*, *Course Class* dan Aturan Kalimat

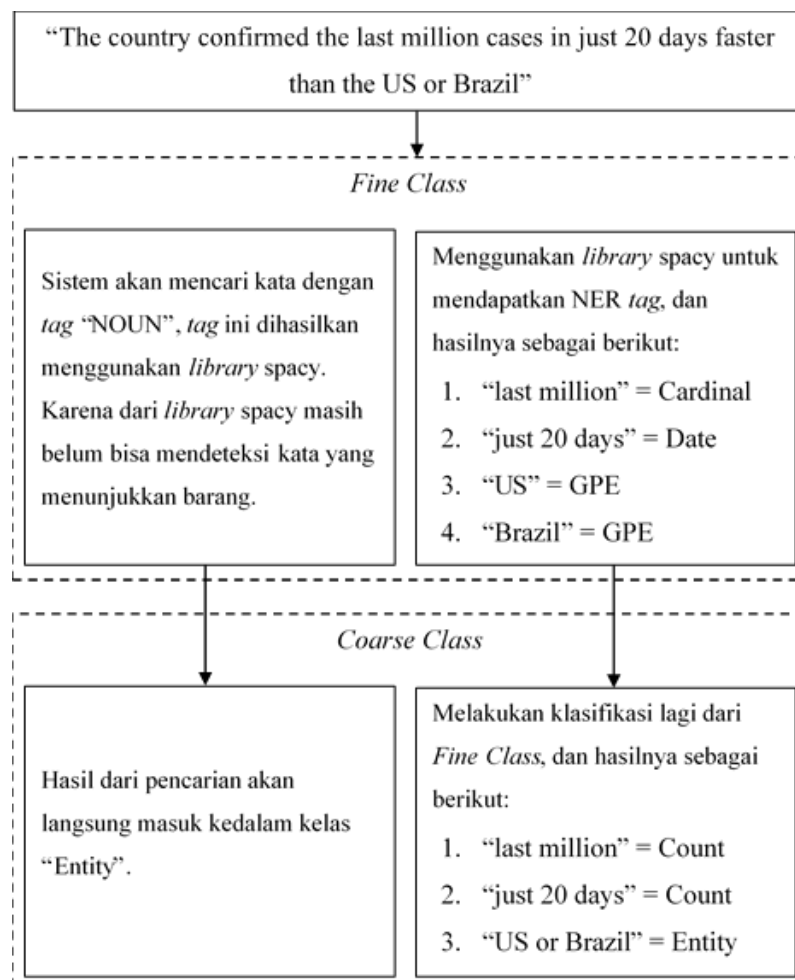
Setelah mendapatkan kalimat sederhana dari tahapan sebelumnya, selanjutnya adalah menentukan *Fine Class* dan *Course Class*, hal ini sesuai dengan penelitian yang dilakukan oleh Ali (2010). Seperti yang terlihat pada Gambar 4.10, *Fine Class* yaitu menentukan kelas awal dari setiap kata pada kalimat, sedangkan *Course Class* adalah pengelompokkan kembali *Fine Class* yang dibagi menjadi lima kelas yang telah ditetapkan, yaitu *Human*, *Location*, *Entitiy*, *Time*, dan *Count*.

Yahya Firdaus, 2020

AUTOMATIC GENERATE QUESTION UNTUK SHORT-ANSWER-QUESTION DI READING COMPREHENSION IELTS MENGGUNAKAN METODE NATURAL LANGUAGE PROCESSING DAN ALGORITMA K-NEAREST NEIGHBOR

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

Penetapan lima kelas ini sesuai dengan penelitian yang dilakukan oleh Ali (2010). Lima kelas inilah yang menjadi aturan kalimat, seperti “*Human verb Human*”, sesuai dengan *coarse class* yang dihasilkan dari kalimat. *Library* yang digunakan pada tahapan ini adalah spacy yang berguna untuk mendapatkan info POS dan NER tag. Gambar 4.10, menunjukkan pada saat mendapat NER tag menggunakan *library*, kata “US” dan “Brazil” terpisah, akan tetapi pada saat pengklasifikasian *Coarse Class* menjadi satu kalimat, hal ini ditentukan oleh sistem yaitu ketika NER tag sama dan diantara kata pertama dan kata kedua terdapat kata yang memiliki tag “CC”, maka kata tersebut akan digabungkan dan menjadi satu kalimat. NER tag untuk kalimat “just 20 days” pun masuk kedalam *Coarse Class* “Count” karena jika dilihat dari strukturnya masuk kedalam “Count”.

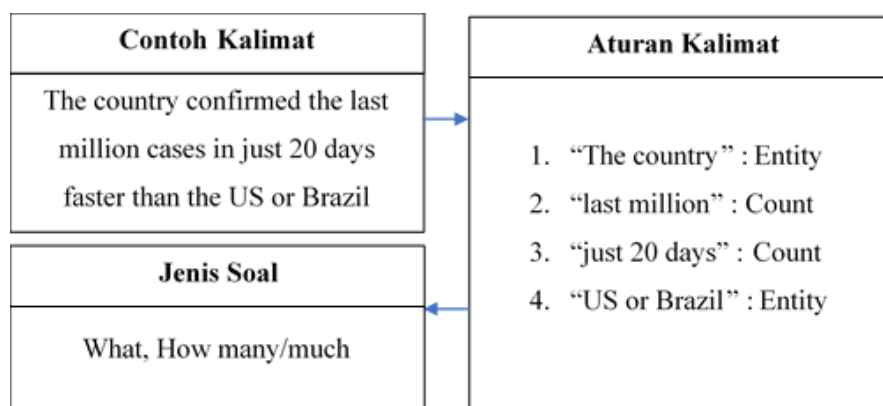


Gambar 4. 10 Contoh pengelompokkan *fine* dan *coarse class* (Ali, 2010).

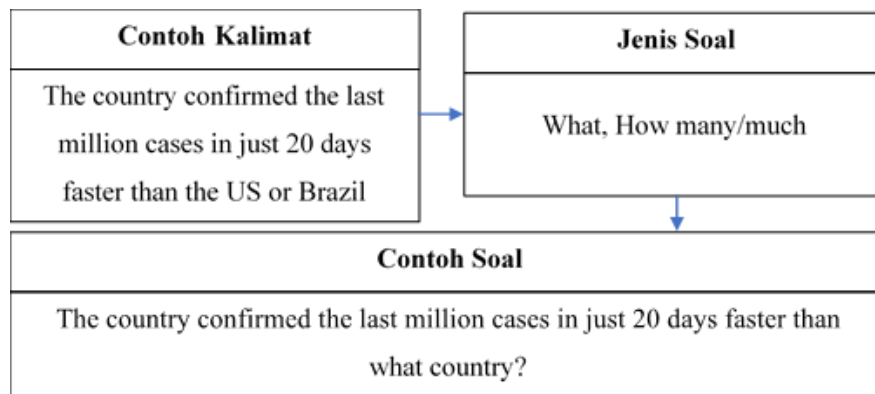
4.2.9 Penentuan Jenis Soal dan Pembuatan Soal

Aturan kalimat hasil dari tahapan sebelumnya akan digunakan pada tahapan ini untuk menentukan jenis soal yang dapat dihasilkan dengan melihat aturan kalimat tersebut, dan pada tahapan ini juga akan menghasilkan kalimat soal serta jawaban yang sesuai dari kalimat, soal tersebut disesuaikan dengan jenis soal. Contohnya pada Gambar 4.11, aturan kalimat akan di ekstrak dari kalimat sederhana, yang kemudian akan didapatkan jenis soal dengan melihat *subject*, *object*, serta *preposition*. Karena pada aturan kalimat memiliki “*time*” maka salah satu jenis soal yang dapat dihasilkan yaitu “*when*”. Jenis soal apa saja yang dapat dihasilkan dengan aturan kalimat seperti pada Gambar 4.11, menggunakan aturan dari penelitian yang dilakukan oleh Ali (2010).

Setelah mendapatkan jenis soal apa saja yang dapat dihasilkan dalam satu kalimat, selanjutnya adalah membuat kalimat pertanyaan sesuai dengan jenis soal. Setiap jenis soal akan menghasilkan satu soal, kecuali pertanyaan dengan jenis “*whom*” yang dalam kalimat terdapat *coarse class* dengan jenis “*Human*” lebih dari satu. Contohnya seperti yang terlihat pada Gambar 4.12, dari jenis soal “*when*” maka akan menghasilkan kalimat soal berupa “*When Husam cook the rice?*”, pertanyaan ini menanyakan waktu atau sesuai dengan aturan kalimat “*Time*”, dengan jawaban yaitu “*at 4 pm*”.



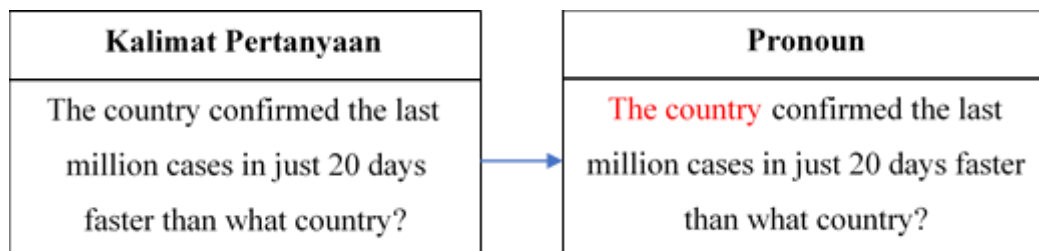
Gambar 4. 11 Contoh saat mendapatkan jenis soal.



Gambar 4. 12 Contoh generate soal dari jenis soal.

4.2.10 Pembersihan dari *Pronoun*

Pada tahapan ini kalimat soal yang dihasilkan pada tahapan sebelumnya yaitu tahapan *generate* soal. Kalimat soal yang dihasilkan tersebut masih harus diproses dengan memilah atau membersihkan kalimat-kalimat soal dan jawaban dari *pronoun*, karena dengan adanya kalimat-kalimat soal dan juga jawaban yang memiliki *pronoun* maka akan menimbulkan ambigu dalam menjawab soal tersebut. Oleh karena itu pada tahapan ini akan menghapus kandidat soal yang memiliki *pronoun* di kalimat soal maupun di jawabannya. Contohnya dapat dilihat pada Gambar 4.13, pada kalimat pertanyaan tersebut memiliki *pronoun* yaitu “The country”, oleh karena itu pertanyaan tersebut akan dihapus dan tidak akan diproses pada tahapan selanjutnya.

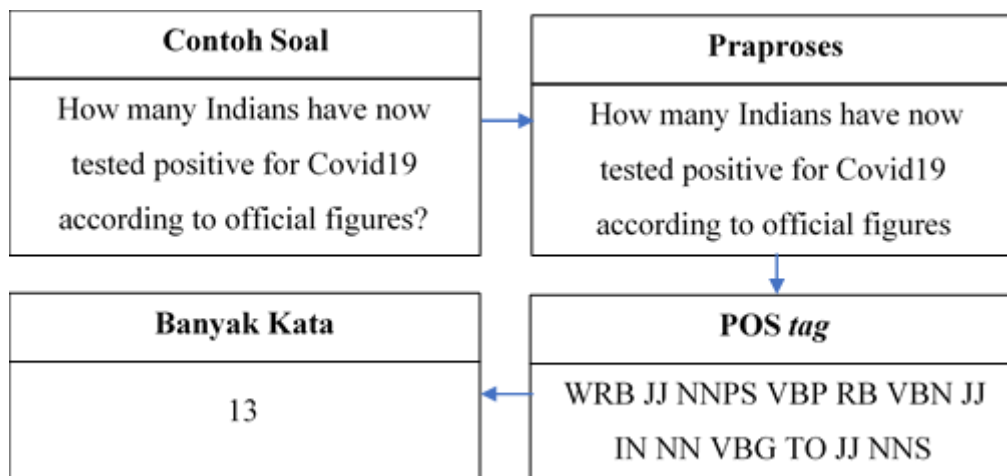


Gambar 4. 13 Contoh kalimat soal yang memiliki *pronoun*.

4.2.11 Praproses dan POS *Tag* dari Kandidat Soal

Langkah selanjutnya setelah mendapatkan kandidat soal adalah mengkonversi menjadi POS *tag* dan menentukan banyak kata dalam kalimat kandidat soal, seperti data latih soal yang pada tahapan sebelumnya dilakukan. Data POS *tag* dan banyak kata dalam kalimat soal ini berguna untuk menentukan

kelayakan soal dari kandidat soal yang dihasilkan oleh sistem. Langkah-langkah dari POS *tag* dan penentuan banyak kata dalam kalimat dapat dilihat pada Gambar 4.14. Dari kalimat contoh soal dilakukan praproses untuk menghilangkan karakter dan simbol, setelah itu dikonversi untuk mendapat POS *tag*, yang akhirnya didapatkan banyak kata dalam kalimat dengan menghitung banyaknya POS *tag*. *Library* yang digunakan pada tahapan ini adalah StanfordCoreNLP yang bertujuan untuk mengkonversi kalimat menjadi *tree*.



Gambar 4. 14 Hasil praproses, konversi POS *tag*, dan banyak kata.

4.2.12 Konversi POS *tag* Menjadi Numerik

Sebelum menggunakan rumus k-NN untuk menentukan jarak antara data latih dengan data uji, POS *tag* dikonversi terlebih dahulu menjadi nilai numerik. Karena dalam menentukan jarak tersebut POS *tag* harus dalam bentuk numerik agar memudahkan dalam melakukan perhitungan. Langkah pertama yaitu inialisasi setiap *tag* menjadi sebuah angka, untuk angka dari setiap *tag* dapat dilihat pada Tabel 4.6.

Tabel 4. 6. Inialisasi nilai *tag*.

Tag	Nilai	Tag	Nilai	Tag	Nilai
CC	1	NNS	13	TO	25
CD	2	NNP	14	UH	26
DT	3	NNPS	15	VB	27
EX	4	PDT	16	VBD	28
FW	5	POS	17	VBG	29
IN	6	PRP	18	VBN	30

Tag	Nilai	Tag	Nilai	Tag	Nilai
JJ	7	PRP\$	19	VBP	31
JJR	8	RB	20	VBZ	32
JJS	9	RBR	21	WDT	33
LS	10	RBS	22	WP	34
MD	11	RP	23	WP\$	35
NN	12	SYM	24	WRB	36

Setelah mendapatkan nilai setiap *tag*, selanjutnya adalah menentukan nilai *S*, dengan ketentuan *range* nya yaitu dari 0 sampai 100 dan ada 36 *tag*. Maka untuk perhitungannya dapat dilihat pada Gambar 4.15, rumus yang digunakan sudah dijelaskan pada bab 2, sehingga didapat nilai *S* yaitu 2,86. Karena nilai dari *S* sudah didapat, selanjutnya adalah menghitung nilai *V*, yaitu nilai numerik dari setiap *tag*. Seperti yang terlihat pada Gambar 4.16, untuk mendapatkan nilai *V* ini yaitu nilai *S* dikali dengan nilai *tag* lalu dikurang satu, sehingga didapatkan nilai *V*.

$$S = \frac{100}{36-1} = 2,86$$

Gambar 4. 15 Perhitungan untuk mendapatkan nilai S.

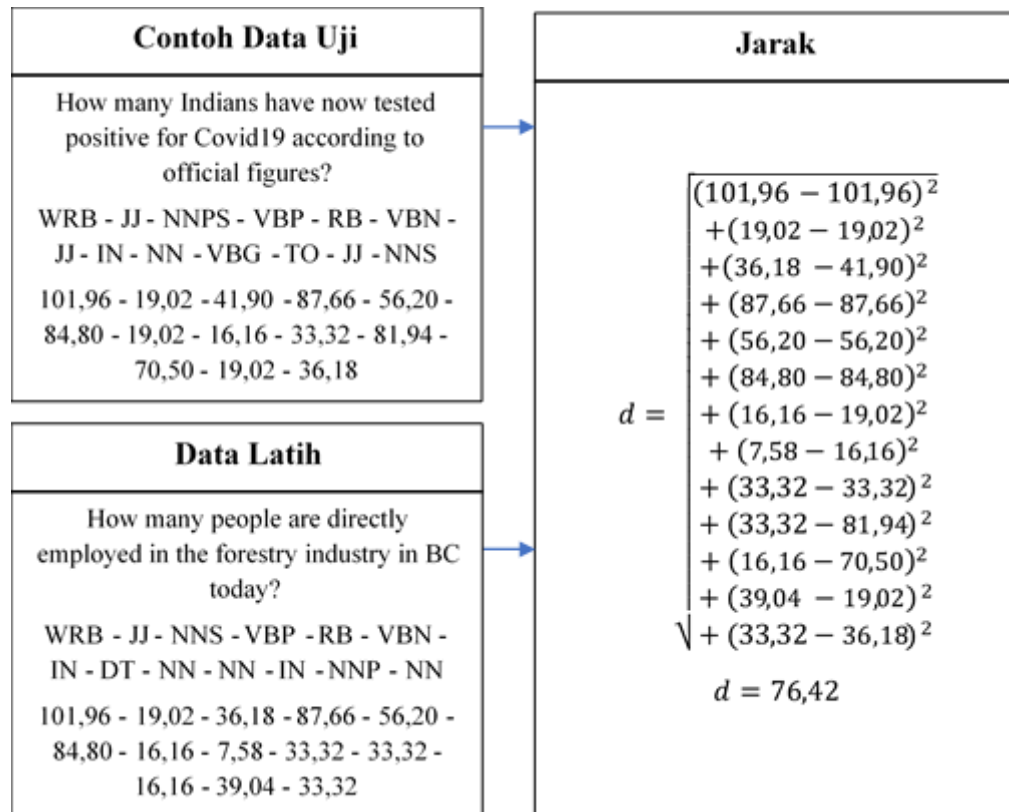
$$\begin{aligned}
 O &= -1, P = \text{Nilai tag} \\
 V &= (2,86 \times P + O), \text{tag} = VB = 27 \\
 V &= (2,86 \times 27 + (-1)) = 76,22
 \end{aligned}$$

Gambar 4. 16 Contoh perhitungan untuk mengubah data kategorik menjadi numerik.

4.2.13 Menentukan Jarak Menggunakan K-NN

Setelah nilai numerik dari POS *tag* didapatkan, maka untuk menghitung jarak antara data uji dengan data latih dapat dilakukan. Rumus yang digunakan yaitu seperti yang telah dijelaskan pada bab 2. Terlihat pada Gambar 4.17, merupakan contoh perhitungan jarak antar data latih dengan data uji, hasil dari perhitungan ini

adalah 91,96. Semakin kecil angka tersebut maka semakin mirip antara data uji dengan data latih, karena hasilnya 91,96 maka dapat dikatakan untuk soal data uji tersebut tidak mirip dengan soal data latih.



Gambar 4. 17 Contoh jarak antara data latih dengan data uji.

4.2.14 Pengurutan Kalimat Soal

Setelah mendapatkan *list* dari kalimat soal yang sudah dipilah dari kalimat-kalimat soal dan juga jawaban yang memiliki *pronoun* dan mendapatkan nilai jarak antara data *training* dengan data uji, maka selanjutnya adalah melakukan pengurutan kalimat soal sesuai dengan jaraknya dengan data *training*. Agar didapatkan *list* kandidat soal yang berurutan sesuai dengan jarak kedekatannya.

4.2.15 Mengubah Menjadi Sinonim

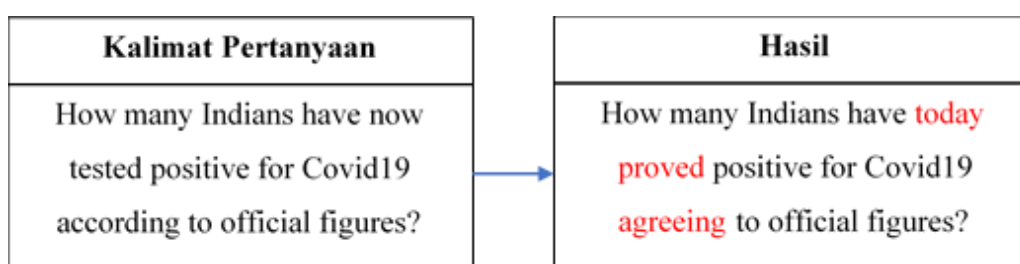
Selanjutnya yaitu tahapan pengubahan beberapa kata yang sudah ditentukan dengan sinonimnya. Kata-kata dalam kalimat yang akan diganti dengan sinonimnya antara lain kata yang memiliki *tag adjective*, *adverb*, dan juga *verb*. Pengubahan kata-kata tertentu menjadi sinonimnya ini bertujuan agar kalimat soal yang dihasilkan akan menjadi lebih sulit dibandingkan dengan kalimat aslinya yang kata-

Yahya Firdaus, 2020

AUTOMATIC GENERATE QUESTION UNTUK SHORT-ANSWER-QUESTION DI READING COMPREHENSION IELTS MENGGUNAKAN METODE NATURAL LANGUAGE PROCESSING DAN ALGORITMA K-NEAREST NEIGHBOR

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

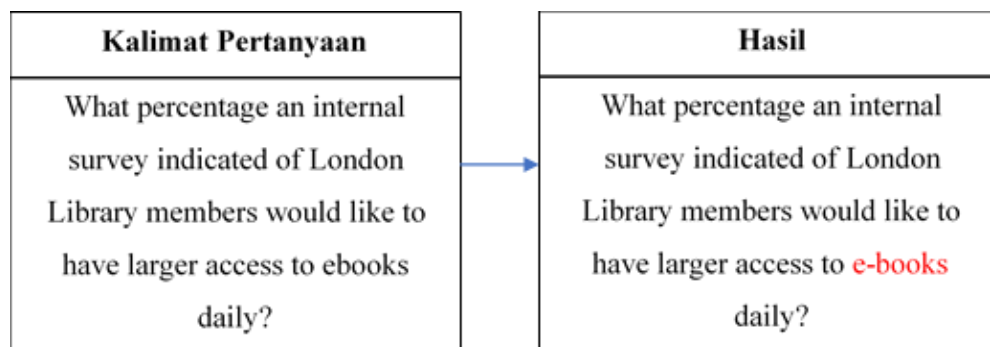
katanya diambil dari teks artikel. Oleh karena itu tahapan ini diperlukan agar kandidat soal yang dihasilkan memiliki tingkat kesulitan yang lebih sulit daripada sebelumnya. Contoh pengubahannya dapat dilihat pada Gambar 4.18, dapat dilihat bahwa kata “now” diubah menjadi sinonimnya yaitu “today”, dan kata “tested” menjadi “proved”. Tentu saja dalam perubahan ini diperhatikan juga jenis dari *verb*, *adjective*, dan juga *adverb* nya. *Library* yang digunakan pada tahapan ini adalah WordNet yang digunakan untuk mendapatkan sinonim, lalu *pyinflect* yang digunakan untuk menyamakan jenis POS *tag*, dan kemudian Word2Vec yang digunakan untuk menentukan jarak antara kata asli dengan sinonimnya.



Gambar 4. 18 Contoh perubahan kata tertentu menjadi sinonimnya.

4.2.16 Pengecekan *Grammar*

Tahapan ini adalah melakukan pengecekan *grammar* dari kalimat kandidat soal, tahapan ini bertujuan untuk melakukan koreksi pada kesalahan *grammar* yang ada pada kalimat kandidat soal yang kemudian akan menghasilkan kalimat kandidat soal dengan kesalahan *grammar* yang sedikit. Melakukan pengecekan *grammar* ini menggunakan *library* yang ada pada bahasa pemrograman python yaitu *language tool*. Dalam melakukan pengecekan *grammar* dapat dilakukan menggunakan *library* *language_check* maupun *language_tool*, akan tetapi penulis memilih *language_tool* karena setelah melakukan perbandingan, *language_tool* lebih spesifik dalam melakukan pengecekan karena sampai karakter tulisan diperhatikan. Contohnya seperti yang terlihat pada Gambar 4.19, pada Gambar 4.19 merupakan hasil pengecekan menggunakan *library* *language_tool*. Perubahannya yaitu yang awalnya “ebooks” diubah menjadi “e-books”.



Gambar 4. 19 Contoh pengecekan *grammar* dan perbaikan.

4.2.17 Pembersihan dari Duplikasi

Tahapan ini merupakan tahapan selanjutnya setelah melakukan pengecekan *grammar*, yaitu melakukan pemilahan atau membuang kandidat soal yang berasal dari kalimat yang sama, sehingga kalimat soal yang dihasilkan sama, hanya saja berbeda dalam hal jawabannya. Tujuannya yaitu agar menghilangkan kandidat soal dari kalimat soal yang sama, karena jika dari kalimat yang sama maka jawaban dari soal tersebut akan ada dalam kalimat soal yang lain.

4.2.18 Pemilihan Soal Dari Kandidat Soal

Pada tahapan ini yaitu menentukan kandidat soal mana yang akan ditampilkan sebagai soal yang layak. Soal yang layak akan ditampilkan dengan jarak terdekat pada data latih. Sedangkan untuk banyak kandidat soal yang ditampilkan tergantung kepada inputan atau permintaan dari *user*. Kandidat soal yang ditampilkan ini yaitu soal dengan jarak terdekat dengan data latih, sehingga kelayakan untuk kandidat soal menjadi soal lebih sesuai dengan soal jenis *short answer question* bagian *reading comprehension* IELTS. Akan tetapi kandidat soal yang ditampilkan disesuaikan maksimalnya dengan banyaknya kandidat soal yang dihasilkan sampai proses yang terakhir. Contoh hasil akhir dari proses penghasilan soal secara otomatis ini dapat dilihat pada Gambar 4.20, yang mana *template* nya disesuaikan dengan *template* jenis soal *short answer question* di *reading comprehension* IELTS.

More than two million Indians have now tested positive for Covid-19, according to official figures. The country confirmed the last million cases in just 20 days, faster than the US or Brazil which have higher numbers. Testing has been expanded considerably in India in recent weeks but the situation varies across states. Spurred by a low death rate, the nation continues to reopen even as new hotspots drive the surge in cases. But some states have imposed restrictions. The recent measures include local, intermittent lockdowns, sometimes limiting activity in specific cities or districts. India is now the third country to cross the two million mark. It reported 62,170 cases in the past 24 hours, taking its total tally up to 2,025,409. The country has reported around 40,700 deaths so far. While that is the world's fifth-biggest total, experts say it is not very high given the country's population of 1.3 billion.

Questions 1 – 2

Answer the questions below.

*Choose **NO MORE THAN THREE WORDS AND/OR A NUMBER** from the text for each answer*

Write your answers in boxes 1-2 on your answer sheet

1. What country is today the 3rd country after crossing the two million mark?
2. How many Indians have today proved positive for Covid19 agreeing to official figures?

Answers

1. India
2. More than two million / More than 2.000.000

Gambar 4. 20 Contoh paragraf dan soal hasil *generate* sistem.

4.3 Pengembangan Perangkat Lunak

Pada subbab ini akan dijelaskan tahapan rekayasa perangkat lunak yang dilakukan sesuai dengan alur sistem yang telah dijelaskan pada bab sebelumnya. Pengembangan perangkat lunak dilakukan dengan menggunakan bahasa pemrograman python dengan menggunakan *google colabs* sebagai *notebook* untuk membuat sistem *automatic question generation* oleh penulis. Lebih jelasnya akan dijelaskan dalam subbab sesuai dengan metode yang penulis pakai dalam mengembangkan sistem *automatic question generation*, yaitu metode *waterfall*.

4.3.1 Analisa Kebutuhan

Penelitian ini bertujuan untuk membuat sistem *automatic question generation* yang berguna untuk membuat soal secara otomatis, menghasilkan data tes dengan tipe soal *short answer question* dan disesuaikan dengan data latih jenis soal *short answer question* yang ada pada *reading comprehension* IELTS. Menentukan kelayakan dari soal yang dihasilkan oleh sistem, yaitu dengan di uji kepada *expert judgement* dan diukur menggunakan instrument penelitian yang cocok untuk jenis soal yang dihasilkan sistem ini.

Data yang digunakan dalam penelitian ini yaitu dataset soal dengan jenis *short answer question* dari *reading comprehension* IELTS. Sumber dari soal dan cara melakukan praproses sudah diejaskan pada subbab sebelumnya, dan juga untuk akat serta bahan untuk melakukan penelitian telah dijelaskan pada subbab sebelumnya.

Sistem yang dikembangkan dalam penelitian ini tentunya memiliki keterbatasan, adapun batasan dari sistem pada penelitian ini yaitu diantaranya:

1. Dari beberapa jenis soal yang ada di *reading comprehension* IELTS, sistem yang dikembangkan hanya sesuai untuk jenis soal *short answer question*.
2. Sistem ini dikembangkan menggunakan *google colabs*, dan menggunakan bahasa pemrograman python.
3. Melakukan konversi dari kalimat menjadi *tree* dilakukan secara manual, karena untuk konversi ini menggunakan server java. Sehingga ketika melakukan uji coba terhadap sistem ada beberapa bagian yang dilakukan dengan manual.
4. Mengumpulkan data latih dari *ebook* dilakukan secara manual, karena sistem ini belum memiliki *scrapping* soal dengan jenis *short answer question* dari *reading comprehension* secara otomatis.
5. Agar dapat melakukan konversi menjadi sebuah *tree structure* diharuskan melakukan instalasi *library* nya terlebih dahulu.

4.3.2 Desain

Pada subbab ini akan menjelaskan mengenai arsitektur dari proses operasional perangkat lunak, perancangan fitur perangkat lunak, serta desain antarmuka.

4.3.2.1. Proses Operasional Perangkat Lunak

Bagian ini akan membahas mengenai masukan atau *input* dan keluaran atau *output* yang sesuai untuk sistem yang penulis kembangkan. Lebih jelasnya yaitu sebagai berikut:

1. Masukan (*input*)

Sistem *automatic question generation* dengan jenis soal *short answer question* menerima masukan berupa *dataset* soal IELTS bagian *reading comprehension* dengan jenis soal *short answer question*. *Dataset* soal ini dikumpulkan dari sumber ebook IELTS dan website IELTS yang dijadikan sumber terpercaya oleh penulis, dan juga *dataset* soal ini sebelumnya sudah di praproses terlebih dahulu, dikonversi menjadi *tree* untuk mendapatkan POS tag, serta dihitung banyak kata dalam kalimat soal. Data POS tag ini akan digunakan untuk membandingkan data latih dengan data uji.

2. Keluaran (*output*)

Sedangkan untuk keluaran dari sistem *automatic question generation* dengan jenis soal *short answer question* ini yaitu kalimat-kalimat soal atau kandidat soal. Kandidat soal ini sudah melalui beberapa tahapan, yaitu berawal dari ekstraksi kalimat sederhana dari kalimat kompleks sampai dihasilkannya kandidat soal, jumlah kandidat soal yang dihasilkan dapat diatur oleh *user*.

4.3.2.2. Perancangan Fitur Sistem

Pada tahapan ini penulis akan merancang fitur dari sistem yang akan dibangun. Perancangan ini dilakukan agar dalam pembangunan sistem dapat berjalan secara terstruktur dan menjadi acuan apa saja yang harus didahulukan dalam pembangunan sistem *automatic question generation* ini. Adapun hasil dari sistem ini yaitu kalimat-kalimat soal yang dapat digunakan dalam melakukan evaluasi pembelajaran ataupun tes ujian IELTS bagian *reading comprehension*.

Yahya Firdaus, 2020

AUTOMATIC GENERATE QUESTION UNTUK SHORT-ANSWER-QUESTION DI READING COMPREHENSION IELTS MENGGUNAKAN METODE NATURAL LANGUAGE PROCESSING DAN ALGORITMA K-NEAREST NEIGHBOR

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

Pada Tabel 4.7, menunjukkan fitur-fitur yang terdapat pada sistem, fitur ini yang nantinya dapat digunakan oleh *user*.

Tabel 4. 7. Perancangan fitur perangkat lunak.

No.	Menu	Fitur	Kegunaan
1.	<i>Pronoun Collection</i>	Menambahkan dan menghapus data <i>pronoun</i>	Agar kandidat soal yang dihasilkan lebih terhindar dari kalimat soal dan jawaban yang memiliki <i>pronoun</i> .
2.	<i>Question Collection</i>	Menambahkan dan menghapus dataset pertanyaan	Agar kualitas dari kandidat soal yang dihasilkan lebih mirip dengan pertanyaan IELTS.
3.	<i>Article</i>	Menambahkan dan menghapus data artikel	Pengguna dapat menambahkan dan juga menghapus data artikel yang akan di <i>generate</i> menjadi soal.
4.	<i>Generate Question</i>	Memasukkan jumlah banyak kandidat soal	User dapat memasukkan banyak kandidat soal yang akan ditampilkan.
		Memilih artikel	User dapat memilih artikel yang akan di <i>generate</i> dari data artikel yang sebelumnya sudah ditambahkan.
		Mengunduh hasil <i>generate</i>	User dapat melakukan pengunduhan hasil <i>generate</i> dalam bentuk <i>file</i>

4.3.3 Implementasi

4.3.3.1 Fungsi Sistem

Setelah menentukan desain sistem dan fitur-fitur yang akan ada pada sistem, selanjutnya adalah implementasi terhadap desain sistem kedalam sistem yang akan dikembangkan. Dari pengembangan sistem ini menghasilkan fungsi-fungsi yang dapat digunakan oleh *user* dalam menggunakan sistem ini. Seperti yang terlihat pada Tabel 4.8, menjelaskan kebutuhan fungsional ketika *user* masuk untuk menghasilkan soal. Fungsi-fungsi tersebut dapat digunakan untuk mengelola data latih, dan data uji atau kandidat soal. Fungsi yang terlihat pada Tabel 4.5, digunakan oleh *user* untuk menghasilkan soal dengan tipe soal yang telah ditentukan, yaitu jenis soal *short answer question* pada *reading comprehension* IELTS.

Tabel 4. 8. Kebutuhan Fungsional pada Sistem

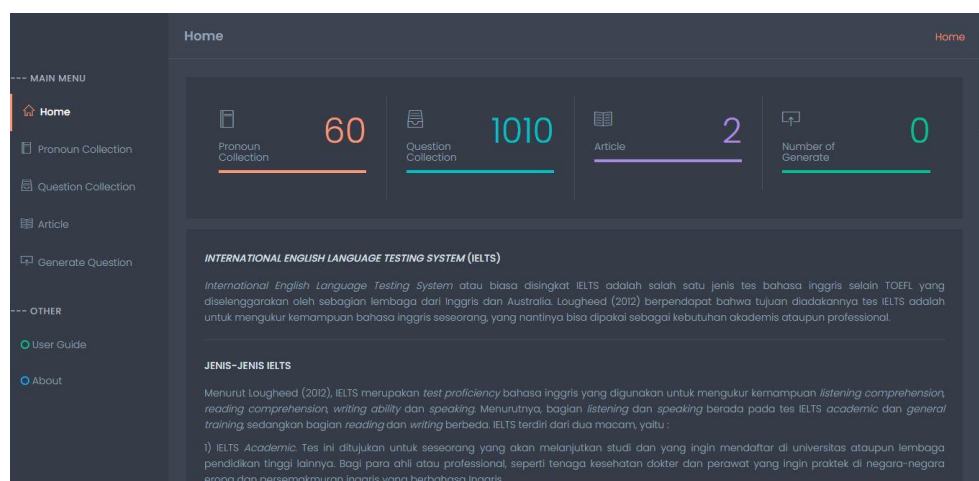
No	Nama Fungsi	Deskripsi
1.	Kelola data <i>pronoun</i>	Digunakan untuk mengelola data <i>pronoun</i>
2.	Kelola Dataset Soal IELTS	Berfungsi untuk mengelola dataset soal
3.	Kelola Artikel	Berfungsi untuk Memasukkan artikel lewat <i>link url</i>
4.	Kelola Banyak Soal	Berfungsi untuk mengatur jumlah banyak soal yang ditampilkan
5.	<i>Generate Question</i>	Berfungsi sebagai penghasil soal otomatis.

4.3.3.2 Implementasi Antarmuka

Dalam pembuatan sistem tentu memerlukan *user interface* atau antarmuka agar sistem dapat digunakan lebih mudah, menjadi jembatan antara pengguna dengan sistem. Semua fungsi atau fitur dari sistem diinterpretasikan dengan lebih menarik dengan adanya antarmuka. Adapun antarmuka dari sistem yang dikembangkan dalam penelitian ini yaitu sebagai berikut:

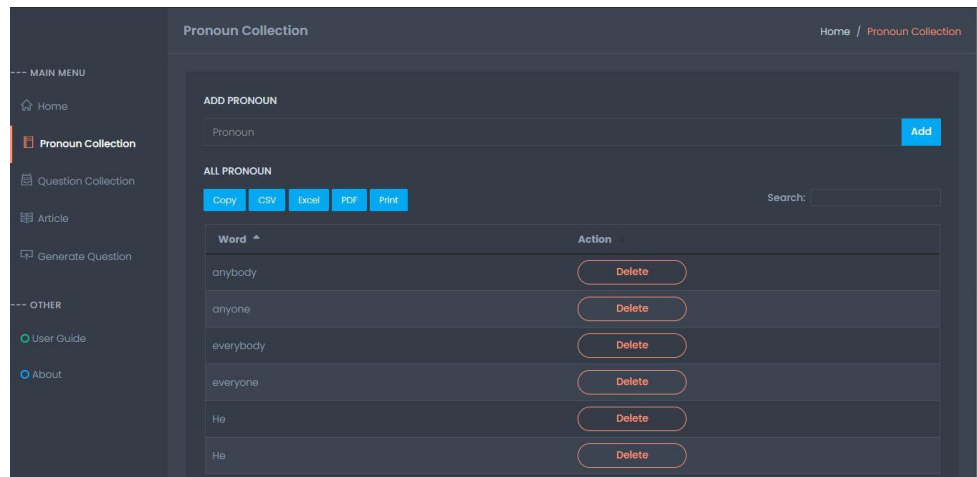
1. Antarmuka *Home*. Saat memulai menjalankan sistem ini, pertama, pengguna akan melihat antarmuka ini, yang mana pada antarmuka *home* akan menyajikan

informasi mengenai pengertian IELTS, jenis tes IELTS, skor IELTS, dan juga mengenai jenis soal *short answer question* pada bagian *reading comprehension* IELTS seperti yang terlihat pada Gambar 4.21, dan juga pada antarmuka *home* menampilkan informasi mengenai jumlah *pronoun*, jumlah dataset pertanyaan, jumlah artikel, dan jumlah berapa kali atau berapa teks yang sudah di *generate* menjadi soal.



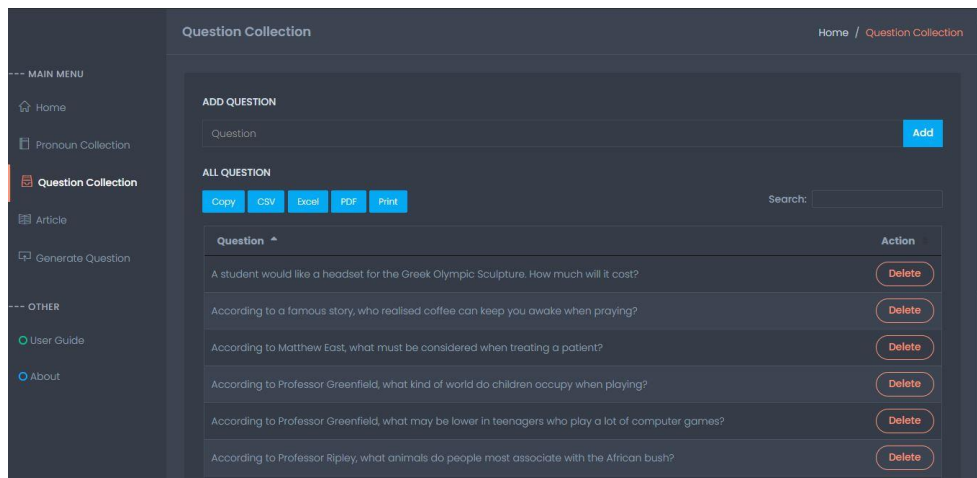
Gambar 4. 21 Antarmuka *home*.

2. Antarmuka *Pronoun Collection*. Pada Gambar 4.22, menunjukkan antarmuka *pronoun collection*, antarmuka ini menyajikan informasi mengenai koleksi *pronoun* yang dimiliki oleh sistem. *Pronoun* ini berguna untuk menghapus kandidat soal yang memiliki *pronoun* pada kalimat soal maupun jawabannya, agar tidak terjadi ambiguitas pada pertanyaannya. Pada antarmuka ini juga pengguna dapat menambahkan ataupun menghapus data *pronoun* agar kandidat soal yang dihasilkan dapat benar-benar bersih dari *pronoun*.



Gambar 4. 22 Antarmuka *pronoun collection*.

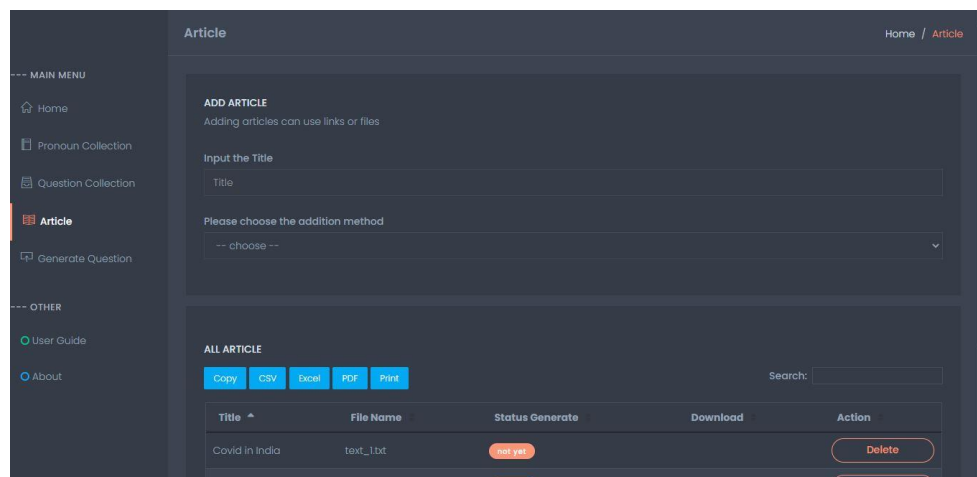
3. Antarmuka *Question Collection*. Antarmuka ini menyajikan informasi mengenai koleksi dataset pertanyaan *short answer question* dibagian *reading comprehension* IELTS. Pengguna pun dapat menambahkan ataupun menghapus data koleksi pertanyaan, hal ini bertujuan agar kualitas soal yang dihasilkan oleh sistem dapat menjadi lebih baik. Antarmuka ini dapat dilihat pada Gambar 4.23.



Gambar 4. 23 Antarmuka *question collection*.

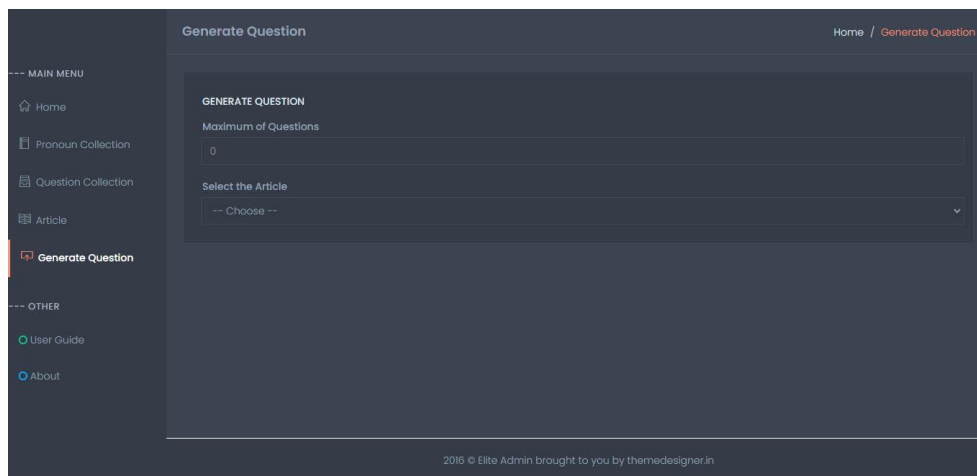
4. Antarmuka *Article*. Sebelum melakukan *generate* soal dari artikel, pengguna diharuskan menambahkan artikel terlebih dahulu pada antarmuka ini, seperti yang terlihat pada Gambar 4.24. Cara untuk menambahkan artikel ini dapat dilakukan dengan dua cara yaitu antara lain yang pertama memasukkan *link* artikel dan yang kedua melakukan *upload file* artikel. Data artikel yang

ditambahkan akan ditampilkan pada tabel dan pengguna pun dapat menghapus data artikel yang ditambahkan.

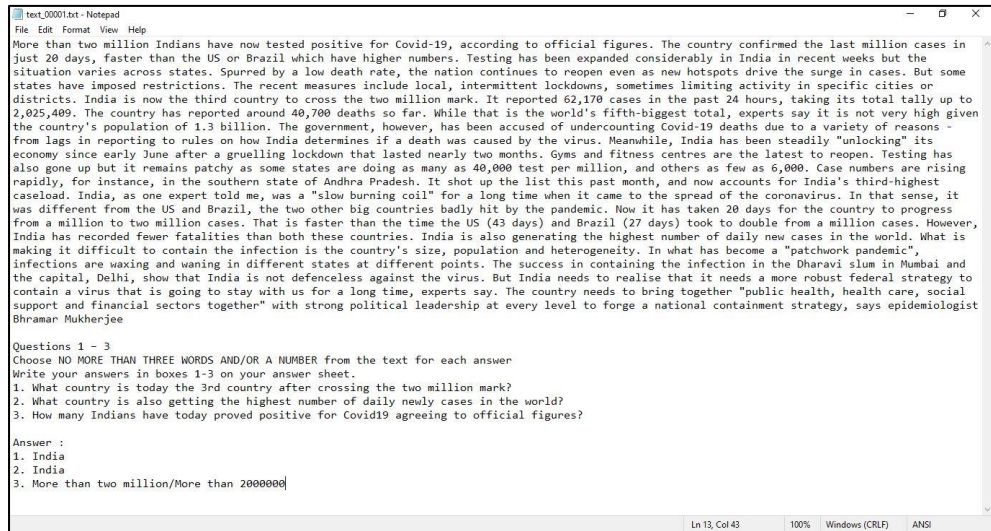


Gambar 4. 24 Antarmuka *article*.

5. Antarmuka *Generate Question*. Pada antarmuka ini pengguna dapat melakukan *generate* soal dari artikel, dengan cara memasukkan jumlah maksimal pertanyaan yang dihasilkan dan juga memilih artikel yang akan di *generate*, seperti yang terlihat pada Gambar 4.25. Hasil dari *generate* akan ditampilkan pada box “*Result*” dan juga pengguna dapat mengunduh *file* hasil *generate*, contohnya dapat dilihat pada Gambar 4.26.

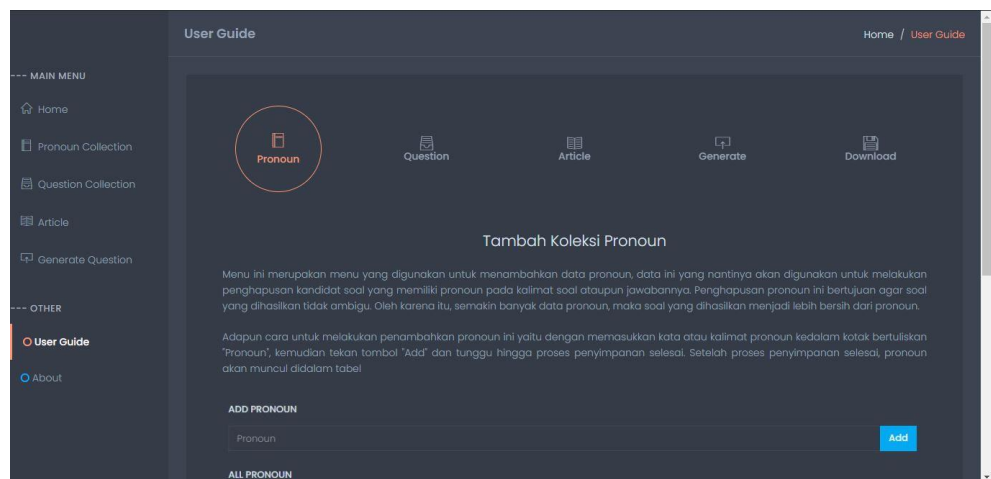


Gambar 4. 25 Antarmuka *generate question*.



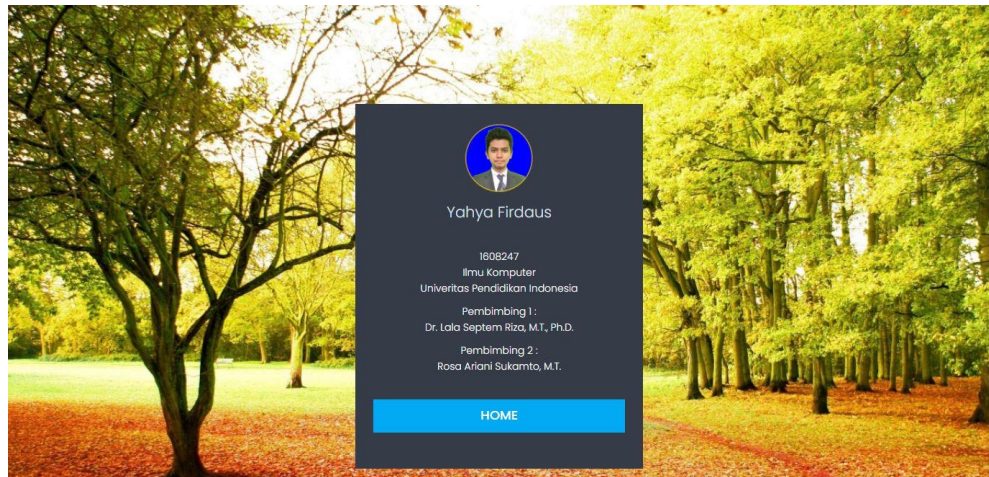
Gambar 4. 26 Hasil akhir *generate* soal dari artikel.

6. Antarmuka *User Guide*. Antarmuka ini bertujuan untuk memberikan petunjuk penggunaan dari sistem ini, seperti yang terlihat pada Gambar 4.27.



Gambar 4. 27 Antarmuka *user guide*.

7. Antarmuka *About*. Antarmuka terakhir dari sistem ini yaitu antarmuka *about* yang menampilkan informasi dari pengembang sistem, seperti yang terlihat pada Gambar 4.28.

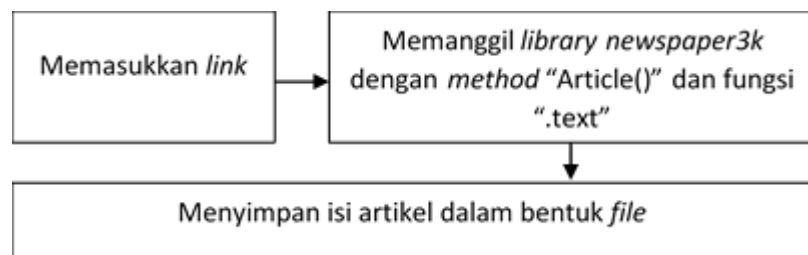


Gambar 4. 28 Antarmuka *about*.

4.3.3.3 Implementasi Kode Program

1. Fungsi *Scraping*

Fungsi ini digunakan untuk mengambil isi artikel dari *link* yang dimasukan oleh *user*. Parameter dari fungsi ini yaitu masukan dalam bentuk *string url*, dari Gambar 4.29, dapat diketahui bahwa untuk melakukan *scraping* ini menggunakan *library newspaper3k*, sebelumnya sudah dilakukan instalasi terlebih dahulu, yaitu dengan *command* “!pip install newspaper3k” pada sel jenis kode di google colabs. Gambar 4.29, menunjukan *flowmap* dari fungsi *scraping*, fungsi ini akan mengambil isi artikel dari *link website* yang dimasukan oleh *user*. *Library newspaper* akan mengambil *author*, *publish date*, dan teks artikel. Teks artikel yang akan diproses lebih lanjut adalah teks artikel yang akan diekstraksi untuk soal-soal nya.



Gambar 4. 29 Flowmap untuk melakukan *scraping*.

2. Fungsi Ekstraksi Kalimat dari Paragraf

Fungsi ini berguna sebagai pengolahan paragraf artikel yang sudah diambil dari *website*. Paragraf ini yang akan diekstraksi menjadi kalimat atau dibagi

Yahya Firdaus, 2020

AUTOMATIC GENERATE QUESTION UNTUK SHORT-ANSWER-QUESTION DI READING COMPREHENSION IELTS MENGGUNAKAN METODE NATURAL LANGUAGE PROCESSING DAN ALGORITMA K-NEAREST NEIGHBOR

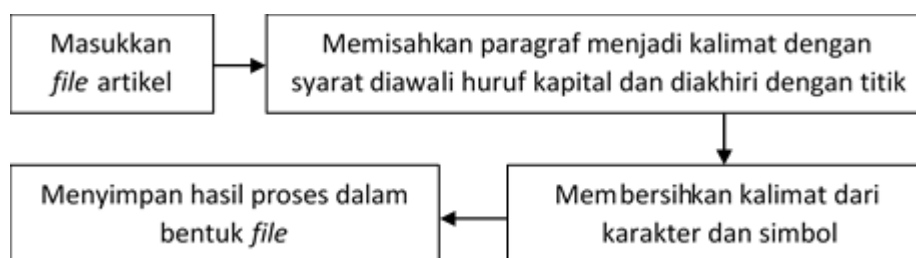
Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

menjadi beberapa kalimat. Syarat agar dapat disebut kalimat yaitu awalan dari kalimat haruslah diawali dengan huruf kapital dan diakhiri dengan titik, syarat ini sesuai dengan pendapat Ali (2010) dalam penelitiannya, contohnya yaitu sebagai berikut:

- a) Tom swims in Park Lake.
- b) 3 women rent an apartment.
- c) tom swims in Park Lake

Terdapat tiga contoh teks yaitu poin a, b, dan c. Dari tiga contoh teks tersebut kalimat yang memenuhi syarat-syarat yaitu hanya poin a, karena diawali dengan huruf kapital dan diakhiri dengan titik. Poin b tidak termasuk kedalam kalimat yang memenuhi syarat-syarat, karena diawali dengan angka walaupun diakhiri dengan titik. Sedangkan poin c tidak memenuhi dari syarat-syarat kalimat karena tidak diawali dengan huruf kapital dan tidak diakhiri dengan titik.

Sedangkan fungsi ekstraksi kalimat dari paragraf ini dapat dilihat pada Gambar 4.30. Pada Gambar 4.30, memperlihatkan *flowmap* untuk fungsi ekstraksi, fungsi ini terlebih dahulu akan mencari kalimat yang diawali dengan huruf kapital dan diakhiri dengan titik. Setelah itu, setiap kalimat yang memenuhi syarat-syarat terkumpul, langkah selanjutnya adalah membersihkan setiap kalimat dari karakter dan simbol, karena untuk ke tahapan selanjutnya haruslah bersih dari karakter dan simbol.



Gambar 4. 30 *Flowmap* untuk praproses dan ekstraksi kalimat.

3. Fungsi Konversi *Tree*

Pada saat melakukan ekstraksi kalimat sederhana dari kalimat kompleks haruslah mendapatkan data *tree structure*, data ini akan digunakan untuk menentukan NP yaitu *Noun Phrase* dan VP yaitu *Verb Phrase*, serta kedalaman dari kata. Fungsi untuk melakukan konversi dari kalimat menjadi *tree* ini terlihat pada

Gambar 4.31, dalam bentuk *flowmap*, untuk melakukan konversi ini penulis

Yahya Firdaus, 2020

AUTOMATIC GENERATE QUESTION UNTUK SHORT-ANSWER-QUESTION DI READING COMPREHENSION IELTS MENGGUNAKAN METODE NATURAL LANGUAGE PROCESSING DAN ALGORITMA K-NEAREST NEIGHBOR

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

menggunakan *library CoreNLPParser* yang disediakan oleh *nlk.parse*. *CoreNLPParser* menggunakan server dari java, hasil dari konversi ini dalam bentuk variabel *tree*, yang kemudian oleh penulis dilakukan praproses untuk mengubah variabel *tree* menjadi menjadi *string*. Hasil dari proses konversi menjadi *tree* ini dalam bentuk *file*.



Gambar 4. 31 Flowmap untuk konversi menjadi *tree*.

4. Fungsi Ekstraksi Kalimat Sederhana

Setelah didapatkan kalimat-kalimat dari paragraf, selanjutnya adalah menentukan apakah kalimat tersebut merupakan kalimat sederhana atau tidak. Syarat kalimat sederhana yaitu harus memiliki satu *subject*, tidak boleh kurang dan tidak boleh lebih, syarat ini sesuai dengan penelitian Kalady, dkk (2010). Gambar 4.32, menampilkan *flowmap* yang digunakan sebagai pemilihan kalimat yang sederhana dan kalimat kompleks. Kalimat sederhana akan masuk ke *list* yang akan masuk ke tahapan selanjutnya, kalimat kompleks akan masuk ke *list* yang akan diproses dan diubah menjadi kalimat sederhana, sedangkan kalimat yang tidak memiliki *subject* tidak akan diproses lebih lanjut.

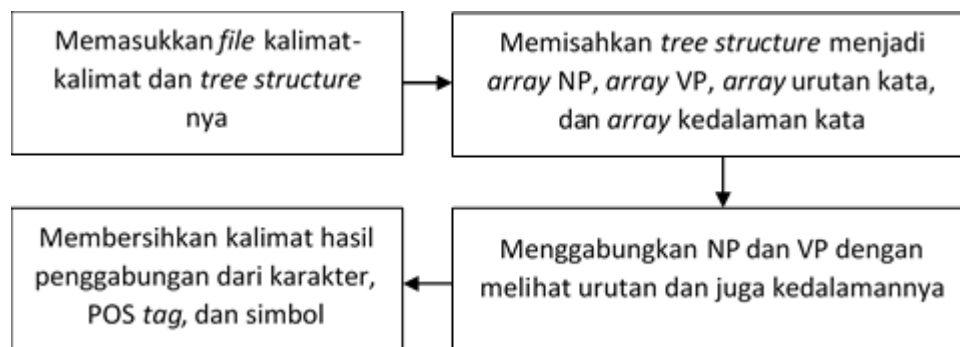
Setelah didapatkan *list* dari kalimat yang masih dalam bentuk kalimat kompleks, selanjutnya adalah mengekstrak kalimat sederhana dari kalimat kompleks yang sudah dikumpulkan, untuk melakukan ekstraksi kalimat sederhana harus mengumpulkan *list* yang diperlukan, yaitu antara lain *list NP*, *list VP*, *list kedalaman pertama*, dan *list urutan kalimat*. Syarat untuk melakukan ekstraksi kalimat ini sesuai dengan penelitian yang dilakukan oleh Ali (2010). Pada Gambar 4.32, menunjukkan *flowmap* yang digunakan untuk mengumpulkan VP atau *verb phrase* yang ada dalam kalimat. Fungsi tersebut akan mengumpulkan VP dilihat dari *tree structure*, dan juga akan memproses untuk mendapatkan *list NP* atau *noun phrase*. NP akan diambil dari lingkungan NP dengan menentukan beberapa kondisi agar NP yang dikumpulkan sesuai dengan yang diharapkan. Pada proses tersebut

Yahya Firdaus, 2020

AUTOMATIC GENERATE QUESTION UNTUK SHORT-ANSWER-QUESTION DI READING COMPREHENSION IELTS MENGGUNAKAN METODE NATURAL LANGUAGE PROCESSING DAN ALGORITMA K-NEAREST NEIGHBOR

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

menghasilkan beberapa *list*, yaitu antara lain *list* NP, *list* VP, kedalaman dari *list* NP dan VP, serta juga urutan dari NP dan VP. Urutan NP dan VP ini digunakan dalam menyusun NP dan VP agar kalimat yang disusun berurutan. Selanjutnya adalah menggabungkan NP dan VP dengan melihat kedalaman dan urutannya, pada Gambar 4.24, juga menunjukkan proses untuk menggabungkan setiap NP dengan setiap VP, lalu kemudian masuk kedalam proses selanjutnya yang mana membersihkan kalimat hasil penggabungan NP dengan VP, karena kalimat hasil ekstraksi masih dalam bentuk *tree*.



Gambar 4. 32 Flowmap ekstraksi kalimat sederhana.

5. Fungsi Klasifikasi

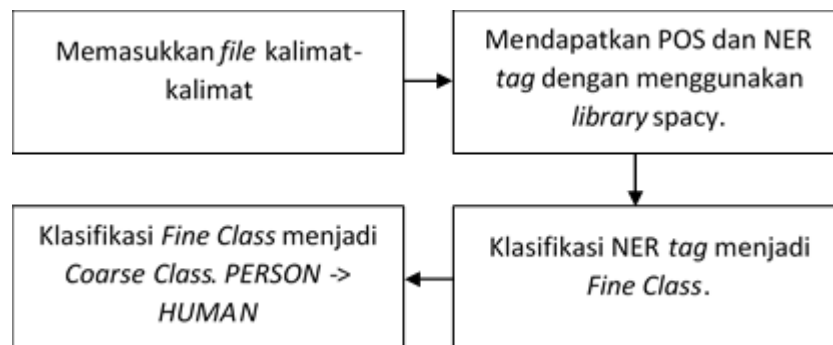
Setelah mendapatkan *list* kalimat sederhana dari sebuah paragraf, selanjutnya adalah melakukan klasifikasi setiap kata dalam kalimat menjadi sesuai dengan peran kata tersebut dalam kalimat. Contohnya kata yang menunjukkan lokasi memiliki peran yaitu “*LOCATION*” dan nama orang memiliki peran yaitu “*PERSON*”. Pada Gambar 4.33, merupakan *flowmap* yang berguna untuk mendapatkan NER *tag* dari kalimat, untuk mendapatkan NER *tag* ini menggunakan *library* spacy, informasi yang didapatkan yaitu antara lain label entitas, teks, dan juga POS *tag*. Hasil NER *tag* ini yang merupakan klasifikasi *Fine Class*.

Tahapan selanjutnya adalah mengubah dari *Fine Class* diklasifikasi kembali menjadi *Coarse Class*. Pengklasifikasian ini sesuai dengan penelitian yang dilakukan oleh Ali (2010), *coarse class* dibagi menjadi lima jenis, yaitu antara lain:

- a) *Human*. Yaitu berupa nama orang atau kata ganti untuk orang itu sendiri, atau yang memiliki NER *tag* “*PERSON*”.
- b) *Entity*. Berupa kata yang mewakili barang, tanaman, hewan, dan objek lainnya.

Sebagai contoh kata “*car*” diklasifikasikan kedalam *entity*.

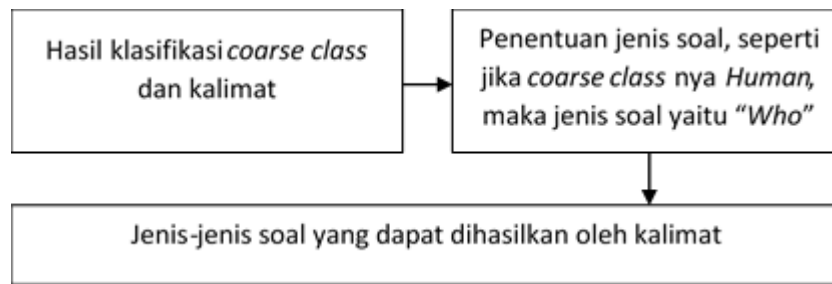
- c) *Location*. Kelas yang mewakili nama tempat dan lokasi, atau kata yang memiliki NER tag “LOCATION”.
- d) *Time*. Kelas ini yang mewakili waktu, tanggal, dan tahun. Kata yang memiliki NER tag “TIME”, dan “DATE” termasuk kedalam kelas ini.
- e) *Count*. Kelas ini mewakili semua kata yang dapat dihitung. NER tag yang masuk kedalam kelas ini yaitu antara lain “PERCENT”, “MONEY”, “QUANTITY”, “ORDINAL”, dan “CARDINAL”.



Gambar 4. 33 Flowmap untuk klasifikasi.

6. Fungsi Penentuan Jenis Soal

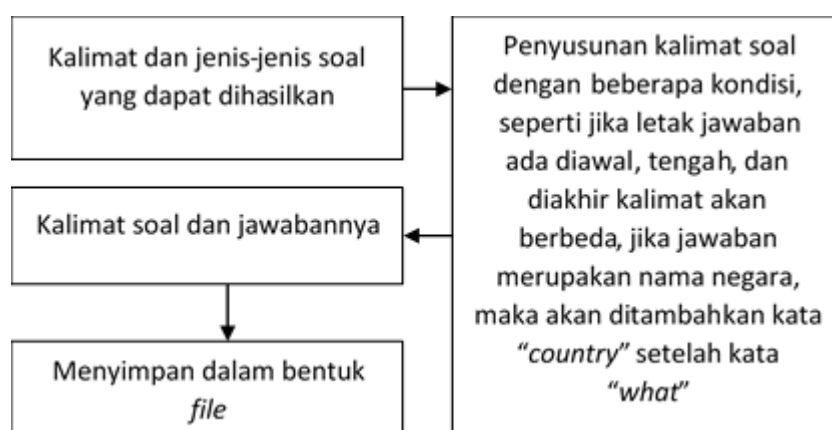
Fungsi penentuan jenis soal berguna untuk mendapatkan jenis soal apa saja yang dapat dihasilkan dalam kalimat sederhana yang sebelumnya sudah diproses, dengan melihat aturan dari kalimat tersebut. Pada Gambar 4.34, menunjukan *flowmap* yang digunakan untuk mendapatkan jenis soal dari aturan kalimat. Penentuan jenis soal ini disesuaikan dengan aturan dari kalimat, atau *coarse class* yang dimiliki pada kalimat, jika pada kalimat terdapat *coarse class HUMAN* maka jenis soal yang dapat dihasilkan adalah “Who”, apabila *coarse class* berupa *ENTITY* maka jenis soal yang dapat dihasilkan adalah “What” atau “Which”, pada kalimat terdapat *coarse class LOCATION* maka jenis soalnya adalah “Where”, jika *coarse class* pada kalimat terdapat *TIME* maka jenis soal yang dapat dihasilkan berupa “When”, dan apabila dalam kalimat terdapat *coarse class COUNT* maka jenis soal yang dapat dihasilkan adalah “How much” atau “How many”.



Gambar 4. 34 Flowmap untuk mendapatkan jenis soal.

7. Fungsi *Generate* Kandidat Soal

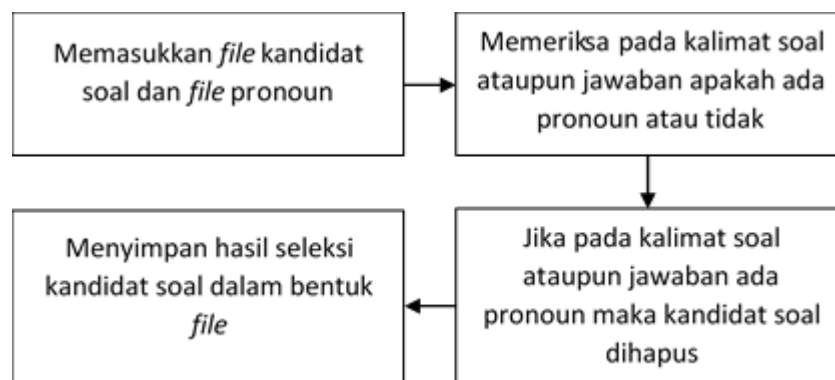
Setelah mendapatkan jenis-jenis soal yang dapat dihasilkan dari kalimat, selanjutnya adalah *generate* kalimat kandidat soal sesuai dengan jenis-jenis soal. Jenis soal yang dihasilkan yaitu *what*, *where*, *when*, *who*, *whom*, dan *how many* atau *how much*. Gambar 4.35, menunjukkan *flowmap* yang menjelaskan proses untuk melakukan *generate* soal dari kalimat sederhana, fungsi tersebut akan melakukan *looping* pada semua jenis soal yang bisa untuk kalimat tersebut, sehingga hasil dari fungsi ini adalah beberapa macam soal dengan jawabannya dari kalimat tersebut. Langkah pertama yang dilakukan dalam proses ini yaitu menyiapkan data kalimat dan juga jenis soal yang dapat dihasilkan dalam bentuk *array*, kemudian selanjutnya adalah melakukan penyusunan kalimat dengan melihat kondisi-kondisi tertentu, contohnya jika jawaban pada kalimat soal terdapat di awal kalimat, maka penempatan kata pertanyaan disimpan diawal, dan jika jawaban merupakan nama negara maka setelah kata “*what*” akan ditambahkan kata “*country*”.



Gambar 4. 35 Flowmap untuk generate kalimat soal.

8. Fungsi Menghapus Kandidat Soal yang Memiliki *Pronoun*

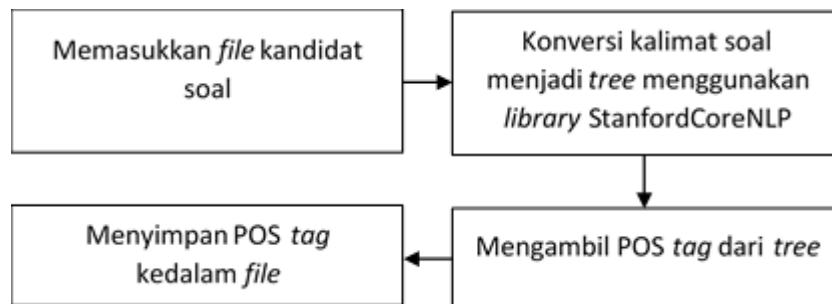
Fungsi ini bertujuan untuk menghapus kalimat-kalimat soal dan juga jawabannya yang memiliki *pronoun*, karena agar kandidat soal yang dihasilkan tidak ambigu dan menjadi pertanyaan kembali. Oleh karena itu fungsi ini digunakan agar menghapus kalimat dan juga jawaban soal yang memiliki *pronoun* agar tidak diproses pada tahapan selanjutnya. Gambar 4.36, menunjukkan *flowmap* yang digunakan untuk melakukan penghapusan atau pemilahan. Pada penentuan *pronoun* ini menggunakan data *pronoun* yang diinputkan dalam bentuk *file txt*, sehingga data *pronoun* dapat dikembangkan atau diperbaharui.



Gambar 4. 36 *Flowmap* untuk menghapus kalimat soal dari pronoun.

9. Fungsi Konversi Kandidat Soal ke POS *Tag*

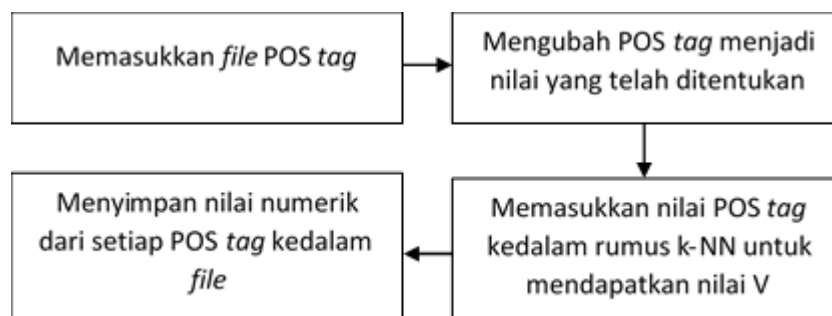
Fungsi ini digunakan untuk melakukan konversi dari kandidat soal menjadi POS tag. Melakukan konversi, sebelumnya dilakukan praproses pada kalimat soal, praproses dilakukan untuk menghilangkan karakter dan simbol dari kalimat soal. Karena untuk melakukan konversi menjadi POS *tag*, diharuskan membersihkan kalimat soal dari karakter dan simbol. Seperti yang terlihat pada Gambar 4.37, sebelum dilakukan konversi menjadi *tree* dan mengambil POS *tag*, terlebih dahulu menghilangkan karakter dan simbol dengan cara me *replace* karakter. Selanjutnya adalah mengkonversi kalimat soal yang sudah dipraproses menjadi *tree*, lalu dilanjutkan dengan mengambil tag per kata dalam *tree*, yang kemudian hasilnya disimpan kedalam *file*.



Gambar 4. 37 Flowmap konversi kandidat soal menjadi POS tag.

10. Fungsi Konversi Tag Menjadi Numerik

Pada Gambar 4.38, menunjukkan *flowmap* dari fungsi yang digunakan untuk melakukan konversi dari tag menjadi numerik, hal ini dilakukan untuk mempermudah dalam menghitung jarak antara data latih dengan data uji, untuk menentukan kelayakan kandidat soal menjadi soal yang layak. Sebelum diubah menjadi numerik, terlebih dahulu menentukan nilai dari tag dan selanjutnya merubah menjadi numerik dengan menggunakan rumus yang sudah dijelaskan pada subbab sebelumnya.

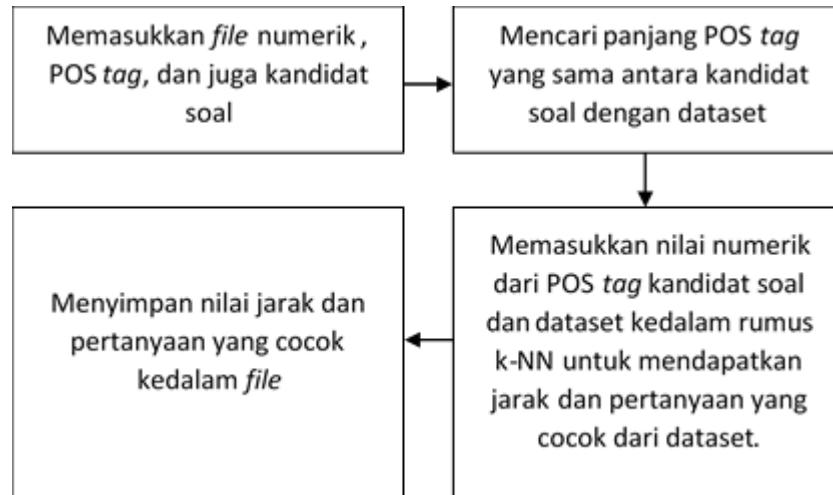


Gambar 4. 38 Flowmap untuk konversi tag menjadi numerik.

11. Fungsi Penentuan Jarak

Tahapan selanjutnya adalah menentukan jarak antara data uji dengan data latih dengan menggunakan nilai numerik. Pada Gambar 4.39, menunjukkan *flowmap* dari fungsi yang digunakan untuk menentukan jarak antara data latih dengan data uji, penentuan jarak ini untuk mengetahui kelayakan dari data uji yang menjadi kandidat soal. Langkah pertama yang dilakukan adalah memasukkan *file* dari nilai numerik, POS tag, dan juga kandidat soalnya. Langkah kedua yaitu mencari pnajang POS tag yang sama antara kadidat soal dengan dataset soal IELTS yang sudah dikumpulkan. Langkah selanjutnya adalah memasukkan nilai numerik dari

setiap POS *tag* kedalam rumus k-NN untuk menghitung jarak antara kandidat soal dengan dataset soal IELTS. Hasil dari proses ini kemudian disimpan dalam bentuk *file*.



Gambar 4. 39 Flowmap untuk menentukan jarak antara data uji dengan data latih.

12. Fungsi Pengurutan Soal

Fungsi ini bertujuan untuk melakukan pengurutan kandidat soal sesuai dengan jarak kandidat soal dengan data *training* nya. Pengurutan ini dilakukan agar mendapatkan kandidat soal dari jarak yang terdekat dari data *training* nya. *Flowmap* dari fungsi pengurutan ini dapat dilihat pada Gambar 4.40, pengurutannya menggunakan *method* yang sudah disediakan oleh python yaitu “*sort()*”.

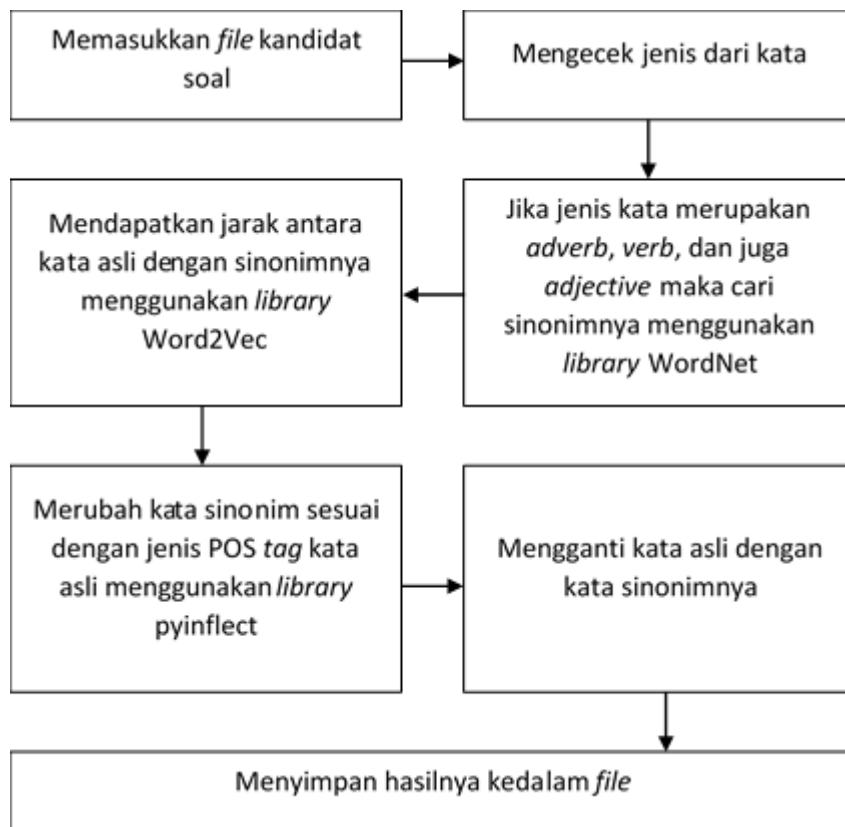


Gambar 4. 40 Flowmap untuk pengurutan soal.

13. Fungsi Pengubahan Sinonim

Fungsi ini digunakan untuk merubah kata-kata yang ada dalam kalimat soal menjadi sinonimnya, hal ini digunakan agar kalimat soal yang dihasilkan memiliki tingkat kesulitan yang lebih sulit dan kata-katanya tidak sesuai dengan kata-kata

dalam kalimat artikel. Kata-kata yang dirubah hanya kata yang memiliki jenis *adjective*, *adverb*, dan *verb*. *Flowmap* dari fungsi ini dapat dilihat pada Gambar 4.41, yaitu dengan menggunakan *library* *pyinflect* yang berguna agar kata yang dirubah memiliki jenis yang sama, contohnya pemakaian jenis *verb* 1 ataupun *verb* 2. Sedangkan untuk menentukan sinonim dari kata nya menggunakan *library* *wordnet* untuk mendapatkan kata sinonimnya dan menggunakan *library* *word2vec* untuk mendapatkan nilai jarak kedekatan dari sinonim dan kata aslinya.



Gambar 4. 41 *Flowmap* untuk pengubahan sinonim.

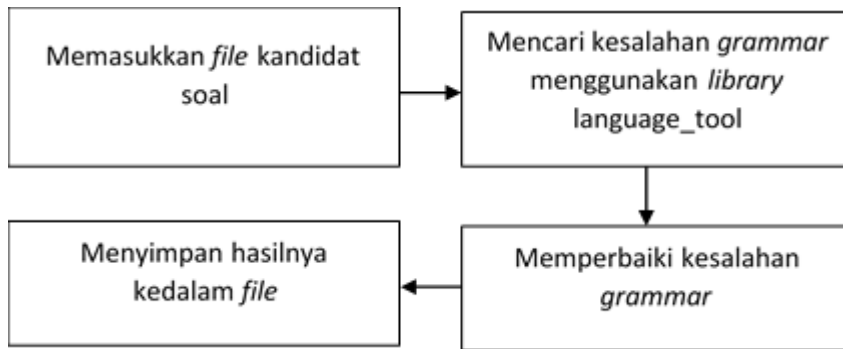
14. Fungsi Pengecekan Grammar

Kalimat soal yang dihasilkan harus dicek *grammar* nya karena dalam pemrosesan kalimat bisa saja terjadi *grammar error*, oleh karena itu fungsi ini digunakan untuk melakukan pengecekan *grammar* dan memperbaikinya. Dalam melakukan pengecekan ini menggunakan *library* *language_tool* dan melakukan perbaikan dari *grammar* yang salah. Seperti yang terlihat pada Gambar 4.42, menunjukkan *flowmap* yang digunakan untuk melakukan pengecekan dan memperbaikinya.

Yahya Firdaus, 2020

AUTOMATIC GENERATE QUESTION UNTUK SHORT-ANSWER-QUESTION DI READING COMPREHENSION IELTS MENGGUNAKAN METODE NATURAL LANGUAGE PROCESSING DAN ALGORITMA K-NEAREST NEIGHBOR

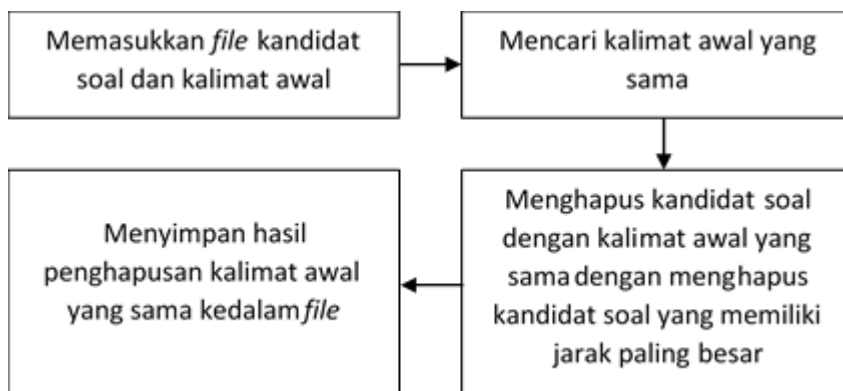
Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu



Gambar 4. 42 *Flowmap untuk pengecekan grammar.*

15. Fungsi Pembersihan Duplikasi

Sedangkan untuk fungsi ini digunakan untuk melakukan penghapusan kalimat soal yang memiliki sumber kalimat yang sama dan hanya mengambil kalimat soal dengan jarak terkecil. Penghapusan duplikasi ini memiliki tujuan agar tidak terjadi dengan contoh jawaban dari soal A muncul pada soal B dan lainnya. *Flowmap* dari fungsi yang digunakan dapat dilihat pada Gambar 4.43, yang mana Langkah pertama adalah memasukkan *file* kandidat soal dan juga kalimat awal dalam menyusun kalimat soal. Langkah selanjutnya adalah mencari kalimat awal yang sama atau muncul beberapa kali, kemudian menghapus kandidat soal yang memiliki kalimat awal sama dengan syarat jarak yang lebih besar yang dihapus. Setelah itu, hasilnya disimpan dalam bentuk *file*.



Gambar 4. 43 *Flowmap untuk pembersihan duplikasi.*

4.3.3.4 Tata Cara Penggunaan Sistem

Tata cara penggunaan dari sistem *automatic question generation* dengan jenis soal *short answer question* pada bagian *reading comprehension* IELTS, yaitu sebagai berikut:

1. Masukan link *url* atau *upload file* artikel yang ingin dilakukan ekstraksi soal dari isi artikel. Menggunakan *scrapping*, maka akan diambil dengan otomatis isi artikel dari *url website* yang dimasukan.
2. Memasukkan banyak kandidat soal yang dihasilkan dan memilih artikel yang akan di *generate* menjadi kandidat soal.
3. Hasil dari *generate* soal akan ditampilkan dan juga pengguna dapat melakukan pengunduhan *file* hasil *generate* dengan format yang disesuaikan dengan jenis soal *short answer question* dibagian *reading comprehension* IELTS.

4.3.4 Pengujian Program

Cara agar mengetahui semua fungsi serta *error handling* telah berjalan sesuai dengan desain alur program, maka penulis menggunakan metode *black box* sebagai tahapan pengujian ini. Tahapan pengujian ini sangat perlu untuk mengetahui apakah program yang dibangun apakah masih memiliki banyak fungsi serta *error handling* yang tidak berjalan sesuai. Hasil dari pengujian ini dapat dilihat pada Tabel 4.8.

Dengan memperhitungkan tentang bagaimana *user* akan menggunakan sistem ini yang dapat menyebabkan pemrosesan tidak berjalan dengan sesuai model alur, penulis dapat menentukan pengujian yang cocok untuk menguji fungsi-fungsi dan *error handling* sistem. Sebagai contoh, *user* melakukan kesalahan pada saat memasukan *tree* saat setelah mendapatkan kalimat pada setiap paragraf di artikel, maka sistem akan menampilkan *error* saat memasukan *tree*.

Pada Tabel 4.9, menunjukan item-item yang penulis ujikan pada tahapan ini, pada sistem yang dikembangkan. Seperti yang terlihat pada Tabel 4.9, bahwa hasil dari pengujian menggunakan metode *black box* sesuai, *error handling* dan berjalan dengan sesuai alur dari model penelitian ini. *Error handling* ini diatur sedemikian rupa sehingga tidak terjadi kesalahan mendasar pada sistem yang dibangun.

Tabel 4. 9. Pengujian *Error Handling* pada Program

No	Item Uji	Hasil yang Diharapkan	Hasil Nyata	Hasil Pengujian
1	Validasi masukan <i>url</i>	<i>Error</i> jika yang dimasukkan bukanlah <i>url link website</i> .	Mengeluarkan peringatan berupa “Maaf, yang anda masukkan bukanlah url website”	Sesuai
2	Validasi masukan jumlah soal	<i>Error</i> jika angka yang dimasukkan kurang dari satu atau melebihi dari jumlah kandidat soal yang dihasilkan oleh sistem.	Sistem akan mengeluarkan peringatan berupa “Maaf, nilai yang anda masukkan melebihi kandidat soal/kurang dari 1”	Sesuai

4.4 Desain Eksperimen

Hal yang pertama kali dilakukan dalam melakukan eksperimen adalah perancangan skenario, dalam hal ini penulis menggunakan tiga skenario, yaitu yang pertama skenario ekstraksi kalimat sederhana dari kalimat kompleks, yang kedua adalah skenario klasifikasi kalimat kandidat soal, dan yang ketiga adalah skenario evaluasi kualitas soal. Melakukan uji kelayakan dari hasil yang dihasilkan ini membutuhkan *expert judgement*, adapun dua *expert* dalam bahasa inggris sebagai *expert judgement* yaitu antara lain:

- a) Edistia Maulida Solihati, S.Pd. selaku lulusan pendidikan bahasa inggris Universitas Pendidikan Indonesia.
- b) Muhammad Rifa’i selaku *freelance* dari tutor di kampung inggris.

4.4.1 Skenario Berita

Skenario pertama ini penulis mengumpulkan artikel-artikel yang akan dipakai untuk menghasilkan kandidat soal dari setiap soal. Tema artikel yang dipakai berbeda setiap artikelnya dengan sumber tiga situs *website* yang berbeda, yang mana dari setiap artikel ini yang nantinya akan menghasilkan kandidat soal, setiap artikel diambil lima soal dengan jarak terendah dengan data latih atau dataset soal IELTS bagian *reading comprehension* dengan jenis soal *short answer question*.

Pada Tabel 4.10 menunjukkan lima sumber artikel yang dipakai dalam penelitian ini. *Website* yang digunakan ada tiga macam, yaitu BBC, CNN, The Jakarta Post, dan New York Times. Ketiga *website* ini ditentukan oleh penulis sebagai sumber terpercaya dengan kesesuaian *grammar* dan lainnya. Tema dari setiap artikel yang digunakan berbeda, yaitu antara lain kesehatan, hoax, liburan, lingkungan, dan olahraga.

Tabel 4. 10. Sumber artikel yang digunakan.

No.	Sumber	Nama Website	Tema
1	https://www.bbc.com/news/world-asia-india-53674857	BBC	Kesehatan
2	https://edition.cnn.com/2020/08/06/world/fake-beirut-video-missile-intl-trnd/index.html	CNN	Hoax
3	https://www.thejakartapost.com/life/2020/10/02/ebook-loans-book-dispensers-how-are-libraries-adjusting-to-the-pandemic.html	The Jakarta Post	Buku
4	https://www.thejakartapost.com/life/2020/10/02/from-athens-to-kyoto-try-coffee-like-a-local.html	The Jakarta Post	Makanan
5	https://www.nytimes.com/2020/10/16/technology/twitter-new-york-post.html	New York Times	Teknologi

4.4.2 Skenario Ekstraksi Kalimat Sederhana

Skenario kedua ini bertujuan untuk mengukur seberapa tepat kalimat sederhana yang dihasilkan dari kalimat kompleks, kalimat kompleks didapat dari ekstraksi setiap paragraf artikel. Penulis melakukan eksperimen ekstraksi kalimat sederhana dari kalimat kompleks. Kalimat sederhana yang dihasilkan dari kalimat kompleks jumlahnya dapat bermacam-macam.

Sistem ini menghasilkan kalimat sederhana sesuai dengan banyaknya NP dan VP yang ada pada kalimat kompleks, juga memperhitungkan kedalaman serta urutan kalimat setiap NP dan VP untuk menghasilkan kalimat sederhana yang susunannya tidak berubah dari kalimat kompleks atau kalimat sebelum dilakukannya ekstraksi. Artikel yang digunakan dalam skenario ini yaitu berasal dari sumber *website* CNN, BBC, The Jakarta Post, dan New York Times, yang kemudian dijadikan penulis sebagai sumber terpercaya dari segi *grammar* serta tata bahasanya. Hasil dari skenario ini berupa kumpulan kalimat sederhana dari kalimat kompleks di paragraf, dan kumpulan kalimat sederhana ini yang nantinya akan diproses lebih lanjut untuk menghasilkan kandidat soal.

4.4.3 Skenario Generate Soal

Pada skenario ketiga ini penulis melakukan *generate* kalimat soal dan kemudian menghasilkan kandidat soal. Tujuan dari skenario ini yaitu melakukan klasifikasi, *generate* kandidat soal dari kalimat sederhana, dan menentukan jarak antara data kandidat soal dengan data latih.

Masukan dari skenario ini adalah kalimat-kalimat sederhana hasil ekstraksi dari kalimat kompleks. Sedangkan hasil dari skenario ini yaitu kandidat soal dengan mengurutkan dari jarak terdekat sampai terjauh dengan data latih. Kumpulan kadnidta soal ini selanjutnya akan diuji kelayakan oleh *expert*, hal ini bertujuan untuk mengetahui seberapa layak soal yang dihasilkan oleh sistem, yang kemudian dilakukannya analisa serta evaluasi pada sistem dan juga soal yang dihasilkan oleh sistem.

4.4.4 Skenario Evaluasi Kandidat Soal

Tahapan selanjutnya setelah menghasilkan kandidat soal adalah melakukan evaluasi kelayakan soal oleh *expert*, peneliti menentukan beberapa syarat untuk *expert* yang akan melakukan evaluasi kandidat soal, yaitu diantaranya *expert* haruslah seseorang yang memang mengambil bidang jurusan bahasa inggris ataupun seseorang yang mengajar bahasa inggris ditempat les. Hal ini bertujuan agar soal yang dihasilkan layak dengan soal *short answer question* di bagian *reading comprehension* IELTS, karena dengan menghasilkan soal yang banyak belum menjadi ukuran untuk kandidat soal yang dihasilkan layak menjadi soal yang dapat dipakai untuk ujian. Oleh karena itu evaluasi diperlukan untuk menentukan kelayakan dari kandidat soal, untuk evaluasi ini penulis menentukan sebuah metrik, yaitu sebagai berikut:

1. *Grammatical Correctness (GC)*

Grammatical correctness, yaitu menentukan apakah kalimat soal yang dihasilkan oleh sistem secara sintaksis terbentuk dengan baik. Poin ini, penulis membagi 3 tingkatan nilai sesuai dengan jumlah *grammar error*, antara lain:

- a) *Best*. Pada kalimat soal tidak memiliki *grammatical errors*.
- b) *Good*. Pada kalimat soal memiliki 1 atau 2 *grammatical errors*.
- c) *Worst*. Pada kalimat soal memiliki 3 atau lebih *grammatical errors*.

2. *Answer Existence (AE)*

Answer existence, yaitu menentukan apakah jawaban dengan pertanyaan sesuai, atau pertanyaan dapat dijawab oleh kunci jawaban dengan benar. Poin ini penulis membagi menjadi 2 tingkatan nilai, antara lain:

- a) *Yes*. Menandakan bahwa jawaban dengan pertanyaan sesuai.
- b) *No*. Menandakan bahwa jawaban dengan pertanyaan tidak sesuai.

3. *Difficulty Index (DI)*

Difficulty index, yaitu menentukan nilai seberapa sulit soal yang dihasilkan oleh sistem. Poin ini penulis membagi menjadi 3 tingkatan nilai, antara lain:

- a) *Easy*. Penilaian ketika pertanyaan yang dihasilkan dianggap mudah.
- b) *Medium*. Penilaian ketika pertanyaan yang dihasilkan dianggap cukup sulit.

c) *Hard*. Penilaian ketika pertanyaan yang dihasilkan dianggap sangat sulit.

Tabel 4.11, menunjukan contoh tabel evaluasi yang akan diserahkan kepada *expert* untuk dilakukan penilaian pada kandidat soal dan data latih. *Expert* akan menilai dari tiga parameter yang telah dijelaskan sebelumnya. Mengitung presentase nilai dari penilaian *expert* menggunakan rumus yang terlihat pada Gambar 4.44.

Tabel 4. 11. Contoh tabel evaluasi oleh *expert*.

Pertanyaan	Grammatical Correctness			Answer Existence		Difficulty Index			Note
	Best	Good	Worst	Yes	No	Easy	Medium	Hard	

$$Persentase = \frac{\text{jumlah skor hasil penilaian}}{\text{skor maksimal}} \times 100\%$$

Gambar 4. 44 Rumus perhitungan persentase penilaian.

4.5 Hasil Eksperimen

Setelah melakukan eksperimen pada sistem *automatic question generation* yang dikembangkan sesuai dengan rancangan, selanjutnya adalah menjelaskan dan menganalisis kandidat soal hasil akhir dari sistem ini.

Eksperimen yang dilakukan membutuhkan soal yang menjadi data latih untuk membandingkan dengan data uji, oleh karena itu pada subbab ini hasil eksperimen akan dibagi menjadi dua yaitu hasil eksperimen ekstraksi kalimat sederhana dan hasil eksperimen *generate* soal.

4.5.1 Hasil Eksperimen Ekstraksi Kalimat Sederhana

Hasil dari eksperimen ini yaitu berupa kalimat-kalimat sederhana yang selanjutnya akan diolah kembali untuk mendapatkan klasifikasi kelas dan menghasilkan kalimat soal dengan acuan dari kalimat sederhana. Langkah pertama yang dilakukan adalah mendapatkan tiap paragraf dari artikel yang kemudian dipisahkan menjadi dalam bentuk kalimat, dengan syarat yang sudah ditentukan.

Yahya Firdaus, 2020

AUTOMATIC GENERATE QUESTION UNTUK SHORT-ANSWER-QUESTION DI READING COMPREHENSION IELTS MENGGUNAKAN METODE NATURAL LANGUAGE PROCESSING DAN ALGORITMA K-NEAREST NEIGHBOR

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

Setelah itu mendapatkan kalimat sederhana dari kalimat kompleks yang ada dalam setiap kalimat hasil pemisahan dari paragraf. Yang terakhir adalah memilih kalimat-kalimat yang sesuai dengan cara menentukan panjang teks, apakah panjang teks sesuai untuk menjadi kalimat soal, apakah panjang dari kalimat sederhana masih dapat dipahami atau tidak dengan cara melihat panjang kalimat sederhana dan kalimat kompleksnya. Pada Tabel 4.12 menampilkan jumlah kalimat, kalimat sederhana, dan kalimat terpilih dari setiap tema artikel yang sudah dikumpulkan.

Tabel 4. 12. Hasil eksperimen ekstraksi kalimat sederhana.

No	Tema	Jumlah Kalimat			
		Awal	Sederhana	Terpilih	Bersih Dari Pronoun
1	Kesehatan	26	40	23	8
2	Hoax	9	22	12	5
3	Buku	22	62	35	8
4	Makanan	33	52	29	10
5	Teknologi	20	48	22	11

4.5.2 Hasil Eksperimen *Generate Soal*

Pada subbab ini, penulis akan menganalisa semua hasil eksperimen *generate* soal dari kalimat sederhana. Setelah mendapatkan kalimat terpilih yang akan digunakan untuk menghasilkan soal dari setiap kalimat, yaitu dengan cara melakukan klasifikasi pada setiap NER *tag* yang terdapat pada kalimat. Yang kemudian NER *tag* tersebut digunakan untuk menjadi bahan pertanyaan. Soal yang dihasilkan antara lain *who*, *whom*, *where*, *when*, *what*, *which*, dan *how many/how much*.

Hasil eksperimen ini merupakan soal dan jawaban yang dihasilkan sistem yang dikembangkan oleh penulis, soal ini yang menjadi kandidat soal untuk soal *reading comprehension* IELTS. Soal yang dihasilkan oleh sistem ini yang nantinya menjadi data uji atau kandidat soal, yang selanjutnya akan dilakukan *expert judgement* untuk mengevaluasi data uji atau kandidat soal. Pada Tabel 4.13, menunjukkan kandidat soal beserta jawaban yang dihasilkan oleh sistem pada eksperimen ini. Sebagai contoh diambil beberapa pertanyaan yang dihasilkan oleh sistem, terdiri dari lima artikel dengan tema berbeda sebagai bahan eksperimen.

Tabel 4. 13. Hasil Eksperimen Generate Soal.

No.	Pertanyaan	Jawaban	Teks
1	What country is today the 3rd country after crossing the two million mark?	India	1
2	What country is also getting the highest number of daily newly cases in the world?	India	1
3	How many Indians have today proved positive for Covid19 agreeing to official figures?	More than two million	1
4	Case numbers are emerging quickly for instance in the southerly state of what?	Andhra Pradesh	1
5	What are already spreading on every Major social media platform?	Doctored videos	2
6	Who is a producer is the Facebook page of Beirut based CNN Arabic social media?	Mehsen Mekhtfe	2
7	In what city fixed videos of Tuesdays devastating explosion are already spreading on every Major social media platform?	Beirut	2
8	What city where the 72 municipal libraries have stayed close for months?	Los Angeles	3
9	What are even limited due to the healthful guidelines in effect in the UK?	London Library	3
10	What percentage an internal survey indicated of London Library members would like to have larger access to e-books daily?	60 percent	3
11	What situation has reinforced the importance of online content at a time when accessing forcible collections?	Coronavirus	3
12	What should so be stewed double in a row?	The coffee	4
13	What is always serviced three times with water brought each time?	The coffee	4
14	What had from brass in The typical Turkish coffee pot foreboded or copper?	the cezve	4
15	Water is so brought to seethe up the potion over what?	a charcoal fire	4
16	The typical what coffee pot is a crucial piece of kit ?	Turkish	4
17	Trump utilizes social media as what?	a megaphone	5
18	The companies prohibited candidates from taking what?	an election victory	5

No.	Pertanyaan	Jawaban	Teks
19	From the start what article was problematic?	the New York Post	5
20	Some people are already utilizing what for call for election violence?	the sites	5
21	What country operatives were discovered to have utilized the sites to seed discord in the 2016 election?	Russian	5

4.6 Analisis dan Evaluasi Hasil Eksperimen

Pada subbab ini penulis melakukan analisa serta evaluasi hasil dari eksperimen yang dilakukan pada sistem ini, baik itu menganalisa dari jarak, *grammar* nya, dan juga keempat parameter dari evaluasi soal yang telah dilakukan oleh *expert judgement*.

4.6.1 Analisis Jarak

Analisis ini dilakukan untuk mengetahui seberapa layak soal yang dihasilkan dengan cara membandingkan antara data uji atau kandidat soal dan data latih, pembandingan ini dilakukan dengan melihat jarak data uji ke data latih. Semakin kecil jarak antara data uji ke data latih, maka semakin sesuai atau layak data uji tersebut, karena data uji tersebut mirip dengan soal-soal IELTS. Pada Tabel 4.14, menampilkan jarak data uji ke data latih.

Tabel 4. 14. Perhitungan Jarak Dengan Data Latih

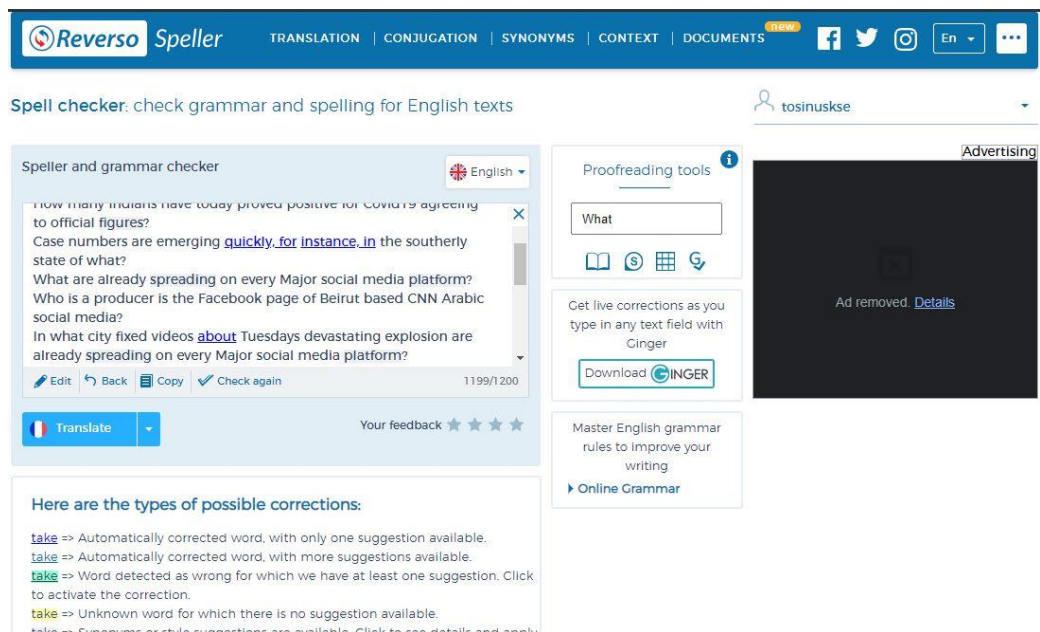
Pertanyaan Contoh	Jawaban	Jarak
What country is today the 3rd country after crossing the two million mark?	India	51.48
What country is also getting the highest number of daily newly cases in the world?	India	65.47
How many Indians have today proved positive for Covid19 agreeing to official figures?	More than two million	76.42

Pertanyaan Contoh	Jawaban	Jarak
Case numbers are emerging quickly for instance in the southerly state of what?	Andhra Pradesh	103.95

Dapat dilihat pada Tabel 4.14, menunjukan kandidat soal beserta jawabannya dan juga jarak kandidat soal dengan data latih. Dapat diketahui bahwa jarak terdekat dari data latih dengan data uji yaitu kandidat soal yang memiliki jarak 51,48. Aturan dari jarak ini adalah semakin kecil nilai jarak yang dimiliki, maka semakin mirip antara data uji dengan data latih, oleh karena itu kandidat soal dengan nilai 51,48 ini merupakan kandidat soal yang paling mirip dengan soal jenis *short answer question* bagian *reading comprehension* IELTS.

4.6.2 Analisis Grammar Soal

Analisis ini digunakan untuk mengetahui kesalahan *grammar* pada kalimat soal, hal ini bertujuan untuk mengetahui apakah kalimat soal yang dihasilkan oleh sistem sudah sesuai dengan aturan *grammar* atau belum. Penulis menggunakan bantuan aplikasi website dalam mengecek *grammar*, yaitu memakai *website* dengan url <https://www.reverso.net/spell-checker/english-spelling-grammar/>. Dengan memakai alat ini dapat dibandingkan hasil dari pengecekan *grammar* yang dilakukan. Tampilan dari *website* yang digunakan untuk melakukan pengecekan *grammar* dapat dilihat pada Gambar 4.45. Cara dari pemakaian *website* ini adalah dengan mengetikkan kalimat soal kedalam box dengan tulisan “Enter or paste your text”, kemudian klik tombol yang bertuliskan “Check”. Kemudian akan diproses dan akan muncul hasil pengecekan. Ketika ada yang salah dalam kalimat, maka akan ada tanda berwarna biru dan juga akan ada penjelasannya dibawah, seperti yang terlihat pada Gambar 4.45.



Gambar 4. 45 Tampilan website pengecekan grammar.

Hasil dari pengecekan grammar ini dapat dilihat pada Tabel 4.15. Pada Tabel 4.15, menampilkan hasil dari pengecekan *grammar* kandidat soal. Angka 0, 1, dan seterusnya ini merupakan jumlah *grammar error* dari kalimat soal. *Grammar error* ini melakukan cek berdasarkan *error*, *misspelling*, *uncertainty*, dan *undefined*. Jumlah kandidat soal dari kelima teks berjumlah 21 soal, dengan setiap artikel diambil maksimal 5 soal dengan jarak terdekat dengan data latih atau data kumpulan soal IELTS bagian *reading comprehension* jenis soal *short answer question*. Dari 21 soal ini, dengan pengecekan *grammar* menggunakan *website* menghasilkan jumlah 5 soal yang memiliki kesalahan *grammar*, sehingga hasilnya adalah presentase *grammar* yang benar sebesar 76,19%.

Tabel 4. 15. Hasil pengecekan grammar.

No.	Pertanyaan	Jumlah Kesalahan	Teks
1	What country is today the 3rd country after crossing the two million mark?	0	1
2	What country is also getting the highest number of daily newly cases in the world?	0	1
3	How many Indians have today proved positive for Covid19 agreeing to official figures?	0	1

Yahya Firdaus, 2020

AUTOMATIC GENERATE QUESTION UNTUK SHORT-ANSWER-QUESTION DI READING COMPREHENSION IELTS MENGGUNAKAN METODE NATURAL LANGUAGE PROCESSING DAN ALGORITMA K-NEAREST NEIGHBOR

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

No.	Pertanyaan	Jumlah Kesalahan	Teks
4	Case numbers are emerging quickly for instance in the southerly state of what?	2	1
5	What are already spreading on every Major social media platform?	0	2
6	Who is a producer is the Facebook page of Beirut based CNN Arabic social media?	0	2
7	In what city fixed videos of Tuesdays devastating explosion are already spreading on every Major social media platform?	1	2
8	What city where the 72 municipal libraries have stayed close for months?	0	3
9	What are even limited due to the healthful guidelines in effect in the UK?	0	3
10	What percentage an internal survey indicated of London Library members would like to have larger access to e-books daily?	1	3
11	What situation has reinforced the importance of online content at a time when accessing forcible collections?	0	3
12	What should so be stewed double in a row?	0	4
13	What is always serviced three times with water brought each time?	0	4
14	What had from brass in The typical Turkish coffee pot foreboded or copper?	0	4
15	Water is so brought to seethe up the potion over what?	1	4
16	The typical what coffee pot is a crucial piece of kit ?	0	4
17	Trump utilizes social media as what?	0	5
18	The companies prohibited candidates from taking what?	0	5
19	From the start what article was problematic?	1	5
20	Some people are already utilizing what for call for election violence?	0	5
21	What country operatives were discovered to have utilized the sites to seed discord in the 2016 election?	0	5
Total		5 Soal	

Yahya Firdaus, 2020

AUTOMATIC GENERATE QUESTION UNTUK SHORT-ANSWER-QUESTION DI READING COMPREHENSION IELTS MENGGUNAKAN METODE NATURAL LANGUAGE PROCESSING DAN ALGORITMA K-NEAREST NEIGHBOR

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

4.6.3 Evaluasi Kualitas Soal

Tahapan selanjutnya setelah menghasilkan kandidat soal adalah melakukan evaluasi kelayakan soal oleh *expert*. Hal ini bertujuan agar soal yang dihasilkan layak dengan soal *short answer question* di bagian *reading comprehension* IELTS, karena dengan menghasilkan soal yang banyak belum menjadi ukuran untuk kandidat soal yang dihasilkan layak menjadi soal yang dapat dipakai untuk ujian. Oleh karena itu evaluasi diperlukan untuk menentukan kelayakan dari kandidat soal. Tabel 4.16 menunjukkan hasil evaluasi yang dilakukan oleh *expert*, dapat disimpulkan bahwa dari *expert* pertama untuk parameter *grammatical correctness* memberikan skor sebesar 29 dengan persentase sebesar 46,03%. Parameter *answer existence* diberikan skor sebesar 39 dengan perhitungan persentase sebesar 92,86%. Sedangkan untuk parameter *difficulty index* diberikan skor sebesar 21 dengan perhitungan persentase sebesar 33,33%. Sedangkan penilaian dari *expert* kedua dengan total skor per parameternya yaitu untuk parameter GC memiliki total 46, persentasenya sebesar 73,02%. Parameter AE memiliki nilai total skor 41 dengan persentase 97,62%. Parameter DI memiliki nilai total skor 23 dengan nilai persentase sebesar 36,51%.

Tabel 4. 16. Hasil evaluasi oleh *expert*.

No. Soal	Expert Pertama			Expert Kedua		
	GC	AE	DI	GC	AE	DI
1	1	2	1	2	2	1
2	2	2	1	3	2	1
3	2	2	1	3	2	1
4	1	2	1	2	2	1
5	3	2	1	3	2	1
6	1	2	1	2	2	2
7	1	1	1	2	1	1
8	1	2	1	2	2	1
9	2	2	1	2	2	1
10	1	2	1	2	2	1
11	1	2	1	3	2	1
12	2	2	1	2	2	1
13	2	2	1	2	2	1
14	1	2	1	2	2	1
15	1	1	1	2	2	1
16	1	1	1	2	2	1

Yahya Firdaus, 2020

AUTOMATIC GENERATE QUESTION UNTUK SHORT-ANSWER-QUESTION DI READING COMPREHENSION IELTS MENGGUNAKAN METODE NATURAL LANGUAGE PROCESSING DAN ALGORITMA K-NEAREST NEIGHBOR

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

No. Soal	Expert Pertama			Expert Kedua		
	GC	AE	DI	GC	AE	DI
17	1	2	1	2	2	1
18	1	2	1	2	2	1
19	2	2	1	2	2	1
20	1	2	1	2	2	2
21	1	2	1	2	2	1
Total	29	39	21	46	41	23
Persentase	46.03 %	92.86 %	33.33 %	73.02%	97.62%	36.51%

Pada Tabel 4.17 menampilkan persentase skor per parameter pada setiap teks. Teks yang dipakai dalam eksperimen terdiri dari 5 jenis tema dengan artikel yang bersumber dari 3 *website* yang berbeda. Dari Tabel 4.17 dapat disimpulkan bahwa penilaian yang dilakukan oleh *expert* pertama untuk parameter GC yang memiliki presentase paling besar adalah teks nomor 2 dengan presentase sebesar 55,56%. Parameter AE yang memiliki presentase paling besar yaitu antara lain teks nomor 1, 3, dan 5 dengan presentase sebesar 100%. Sedangkan untuk parameter DI semua teks memiliki presentase yang sama yaitu 33,33%. Penilaian dari *expert* kedua untuk parameter GC yang memiliki nilai presentase tertinggi sebesar 83,33%, yaitu teks 1. Parameter AE yang memiliki nilai presentase tertinggi ada empat teks yaitu antara lain teks 1, 3, 4, dan teks 5. Sedangkan untuk parameter DI teks yang memiliki nilai presentase paling tinggi yaitu teks 2 dengan nilai presentase sebesar 44,44%.

Tabel 4. 17. Persentase skor pada setiap teks.

No. Teks	Expert Pertama			Expert Kedua		
	GC	AE	DI	GC	AE	DI
1	50.00%	100.00%	33.33%	83.33%	100.00%	33.33%
2	55.56%	83.33%	33.33%	77.78%	83.33%	44.44%
3	41.67%	100.00%	33.33%	75.00%	100.00%	33.33%
4	46.67%	80.00%	33.33%	66.67%	100.00%	33.33%
5	40.00%	100.00%	33.33%	66.67%	100.00%	40.00%

Setelah melakukan perhitungan presentase dengan dilihat dari setiap teks dan setiap *expert*, selanjutnya adalah melakukan perhitungan dengan

menggabungkan hasil evaluasi yang dilakukan oleh *expert* pertama dengan *expert* kedua. Hal ini dilakukan untuk dapat melihat nilai presentase akhir dan juga rata-rata nya. Hasil perhitungan yang telah dilakukan dapat dilihat pada Tabel 4.18, nilai ideal, skor maksimal dari setiap parameter berbeda, yaitu untuk parameter *grammatical correctness* memiliki nilai ideal 3 dengan skor maksimal 63 dan memiliki skor total sebesar 75. Parameter *answer existence* memiliki nilai ideal 2 dengan skor maksimal 42 dan memiliki skor total sebesar 80. Parameter *difficulty index* memiliki nilai ideal 3 dengan skor maksimal 63 dan memiliki skor total sebesar 44.

Selanjutnya adalah melakukan perhitungan persentase dengan memakai rumus yang terlihat pada Gambar 4.44. Hasilnya untuk parameter *grammatical correctness* memiliki presentase sebesar 59,52%, sedangkan parameter *answer existence* memiliki presentase sebesar 95,24%, dan untuk parameter *difficulty index* memiliki presentase sebesar 34,92%. Sehingga rata-rata persentase memiliki nilai sebesar 63,23%. Parameter *answer existence* memiliki skor besar karena soal dengan jawaban memiliki kecocokan yang tinggi, parameter *grammatical correctness* memiliki skor rendah karena pada kalimat soal masih memiliki kesalahan *grammar* yang banyak, dan parameter *difficulty index* memiliki skor yang kecil karena kalimat soal yang dihasilkan masih terbilang mudah, penyusunan kalimat menjadi kalimat soal masih belum kompleks.

Tabel 4. 18. Hasil perhitungan setiap parameter.

Parameter	Nilai Ideal	Skor Maksimal	Skor Total	Persentase	Keterangan
<i>Grammatical Correctness</i>	3	63	75	59.52%	Skor besar maka nilai lebih baik.
<i>Answer Existence</i>	2	42	80	95.24%	
<i>Difficulty Index</i>	3	63	44	34.92%	
Rata-Rata				63.23%	

4.6.4 Analisis Perbandingan dengan Penelitian Sebelumnya

Pada subbab ini peneliti akan melakukan perbandingan antara penelitian ini dengan penelitian sebelumnya, dengan penelitian yang sama yaitu mengenai *automatic question generation*. Karena penelitian mengenai *automatic question generation* itu sendiri sudah banyak dilakukan, dengan metode, *tools*, jenis soal yang dihasilkan, serta hasil yang berbeda-beda. Oleh karena itu analisis mengenai perbandingan dengan penelitian sebelumnya diperlukan, agar dapat membandingkan kelemahan dan juga kelebihan dari penelitian ini dengan penelitian lainnya.

Penelitian ini memiliki jurnal yang menjadi rujukan atau pedoman dalam pengembangan sistem *automatic question generation* dengan jenis soal *short answer question*. Jurnal yang menjadi rujukan tersebut adalah penelitian yang dilakukan oleh Ali (2010). Karena jurnal tersebut menjadi rujukan dalam penelitian ini, maka model alur pengembangan sistem mirip, jenis soal yang dihasilkan pun sama yaitu jenis soal *short answer question*. Walaupun jenis soal yang dihasilkan sama, tetapi jumlah jenis pertanyaan yang dihasilkan berbeda, yaitu pada penelitian ini akan menghasilkan pertanyaan dengan “*which*”, tentunya dengan syarat-syarat yang memenuhi saat pertanyaan tersebut harus memakai “*which*”. Akan tetapi model alur yang dipakai dalam penelitian ini yaitu dalam hal ekstraksi kalimat sederhana dan *generate* soal saja. Dalam penggunaan metodenya pun mirip, akan tetapi *tools* yang digunakan berbeda, yaitu contohnya dalam melakukan konversi dari kalimat menjadi *tree structure*, *library* yang digunakan oleh Ali (2010) adalah *Charniak Parser* sedangkan yang dipakai dalam penelitian ini yaitu menggunakan *StanfordCoreNLP*. Evaluasi yang dilakukan pun berbeda dengan penelitian ini berbeda, penelitian yang dilakukan oleh Ali (2010) menggunakan penilaian *percission* dan juga *recall*. Sedangkan pada penelitian ini menggunakan evaluasi penilaian oleh *expert* dalam bidang bahasa inggris. Nilai *percission* dari penelitian Ali (2010) memiliki nilai sebesar 0,485, sedangkan untuk nilai *recall* sebesar 0,315. Soal yang dihasilkan pun berbeda, karena dalam penelitian oleh Ali (2010) tidak dikhususkan untuk *short answer question* pada bagian *reading comprehension* IELTS.

Penelitian mengenai *automatic question generation* pun sudah ada penelitian skripsi yang sebelumnya, yaitu penelitian yang dilakukan oleh Pertiwi (2018) dan juga Anwar (2018), akan tetapi jenis soal yang dihasilkan dan juga proses pengembangannya berbeda. Penelitian skripsi yang dilakukan oleh Pertiwi (2018) memiliki tujuan menghasilkan soal dengan jenis *sentence completion*, berbeda dengan jenis soal yang dihasilkan oleh sistem yang dikembangkan dalam penelitian ini. Metode yang digunakan yaitu menggunakan *natural language processing* dengan memanfaatkan praproses dan juga *POS tag*, kemudian menggunakan algoritma k-NN untuk menentukan jarak antara data latih dengan data uji. Pemakaian algoritma k-NN ini sama dengan penelitian ini, yaitu untuk menentukan jarak.

Sedangkan penelitian yang dilakukan oleh Anwar (2018) menghasilkan sistem *automatic question generation* dengan jenis soal *error identification*. Pemilihan dari kata yang digaris bawahi untuk menjadi soal pada kalimat menggunakan *levenshtein distance*. Pada penelitian ini juga menggunakan *POS tag* dan juga praproses dalam memproses kalimat. Dari kedua penelitian skripsi sebelumnya tersebut, mereka menggunakan pilihan pengecoh. Dalam melakukan evaluasi, penelitian ini sama menggunakan *expert* dengan parameter yang sama, tapi dalam penelitian ini tidak ada parameter *distractor quality* karena dalam penelitian ini tidak memakai pengecoh, sesuai dengan *short answer question* pada bagian *reading comprehension* IELTS.

Perbedaan penelitian ini dengan kedua penelitian skripsi yang dilakukan oleh Pertiwi (2018) dan Anwar (2018) sangat jelas pada jenis soal, akan tetapi metode dalam menggunakan *POS tag* dan juga praproses yang digunakan sama. Terutama dengan penelitian yang dilakukan oleh Pertiwi (2018) memiliki kesamaan dalam memakai algoritma k-NN untuk menentukan jarak antara data latih dengan data uji, agar dapat mengetahui kelayakan dari soal yang dihasilkan oleh sistem. Hasil dari evaluasi yang dilakukan pun berbeda, yaitu untuk penelitian yang dilakukan oleh Pertiwi (2018) memiliki presentase rata-rata sebesar 81,93%, penelitian yang dilakukan oleh Anwar (2018) memiliki presentase rata-rata sebesar 82%, sedangkan untuk penelitian ini memiliki presentase rata-rata sebesar 63,23%. Penelitian ini memiliki presentase rata-rata yang lebih kecil daripada penelitian

sebelumnya karena dalam penelitian ini melakukan penyusunan kalimat soal dari kalimat aslinya, sehingga dari kompleksitas penyusunan kalimat pada penelitian ini masih kurang.

Tabel 4. 19. Perbandingan dengan penelitian sebelumnya.

Sitasi	Tujuan	Metode	Tools	Hasil	Jenis Soal
Mazidi & Tarau (2016)	Penulis mengusulkan pendekatan baru terhadap sistem penghasil soal otomatis yang dapat meningkatkan presentasi soal diterima dibandingkan dengan sistem state-of-the-art.	NLP, DeconStructure algorithm, SRL	MTurk, Python, Microsoft Research's SPLAT, H&S system	Dari 200 soal dengan membandingkan sistemnya dengan sistem H&S, menghasilkan average rating melebihi Heilman & Smith system, dengan nilai 3,7 dan sedangkan Heilman & Smith system memiliki average rating 2,9. Atau 72% pertanyaan dari hasil generate sistem dapat diterima.	Multiple Choice
Majumder & Saha (2014)	Soal dengan jenis <i>multiple choice</i> merupakan salah satu jenis soal yang digunakan untuk mengukur kemampuan seseorang dalam memahami pembelajaran. Akan tetapi dalam pembuatan <i>multiple choice</i> secara manual sangat membuang waktu. Oleh karena itu peneliti mengembangkan sistem penghasil soal otomatis.	Support Vector Machine (SVM) classifier, Parse Tree Matching (PTM) Algorithm, Conditional Random Field (CRF)	WordNet, Stanford CoreNLP Suit, NER system	Menghasilkan sebuah sistem penghasil soal pilihan ganda dengan akurasi sebesar 93,684% dari 5 evaluator. Dari 95 <i>sentences</i> , evaluator 1 memberikan 90, evaluator 2 memberikan 89, evaluator 3 memberikan 87, evaluator 4 memberikan 90, dan evaluator 5 memberikan 89.	Multiple Choice

Sitasi	Tujuan	Metode	Tools	Hasil	Jenis Soal
Alsubait, et al (2012)	Penilaian kemampuan siswa dalam proses pembelajaran lebih bagus memakai sistem evaluasi atau tes. Dilain hal perancangan dan perencanaan evaluasi membutuhkan waktu yang tidak sedikit. Oleh karena itu penelitian ini dilakukan.	Structure Mapping Theory (SMT), Vector Space Model (VSM)	Gene Ontology, People & Pets Ontology, Pizza Ontology	Menghasilkan pendekatan baru untuk <i>generate multiple choice</i> . Maximum akurasi mencapai 88% dari <i>analogy-solver</i> .	Multiple Choice
Heilman & Eskenazi (2007)	Tujuan dari penelitian ini adalah membuat penghasil soal otomatis dengan tipe spesifiknya adalah <i>vocabulary assessment</i> . Membuat sistem yang dapat menjadi tutor pembelajaran bahasa inggris.	thesaurus extraction technique	WordNet, Gigaword corpus	Penelitian ini menghasilkan sistem penghasil soal otomatis dengan membandingkan dua teknik, yaitu thesaurus extraction technique dan dengan menggunakan manusia. Hasilnya adalah dengan menggunakan sistem mendapatkan presisi 80% dari soal yang dihasilkan.	Multiple Choice
Tsumori & Kaijiri (2007)	Metode yang dipakai untuk generate soal pilihan ganda diterapkan untuk membuat level pemahaman siswa secara otomatis.	technical-term database, question-template database	CAI (Computer Assisted Instruction)	Hasilnya adalah metode dapat diterapkan untuk membuat level pemahaman siswa secara otomatis, walaupun masih ada masalah yang belum terpecahkan karena harus melihat situasinya secara langsung.	Multiple Choice

Sitasi	Tujuan	Metode	Tools	Hasil	Jenis Soal
Kumar, et al (2015)	Melanjutkan pembuatan soal yang bervariasi dari penelitian sebelumnya, yang berguna sebagai metode yang efektif untuk membantu siswa dalam belajar agar lebih baik.	Part-of-Speech (POS) Tagger, Human Intelligence Tasks (HITs), Support Vector Machine (SVM)	Amazon Mechanical Turk, Word2Vec, WordNet, TF-IDF search engine, phyton	Menghasilkan sistem pembuat soal otomatis yang bernama RevUP. Dengan langkah pertama adalah sentence selection, kemudian gap selection, dan yang terakhir adalah distractor selection. Dengan menggunakan discriminative classifier dalam melakukan training, menghasilkan akurasi sebesar 81%. Dengan 94% distractor yang di generate adalah bagus.	Gap Fill
Huang, et al (2012)	Membuat sistem penghasil soal otomatis yang dapat berguna untuk melakukan tes dan evaluasi dari kemampuan berbahasa siswa. Dengan soal pilihan ganda dan memiliki level serta kategori soal yang berbeda.	exponential moving average (EMA), Chi-Square test	Tgrep2 patterns. Sumber teks: Time For Kids, Student Times, Voice of America, CNN, China Post Online and Yahoo! News.	Menghasilkan sistem penghasil soal otomatis dengan jenis soal pilihan ganda dan memiliki level serta kategori soal yang berbeda. Hasil penelitian menunjukan tingkat <i>rectification</i> pada <i>experimental group</i> rata-rata secara signifikan lebih tinggi daripada <i>control group</i> . <i>Experimental group</i> (M=0.54, SD=0.29).	Multiple Choice Gap Fill

Sitasi	Tujuan	Metode	Tools	Hasil	Jenis Soal
Goto, et al (2010)	Membantu siswa dalam belajar dengan membuat sistem penghasil soal otomatis dari berbagai teks, misalkan dari artikel, paper, dan website yang bebas dipilih oleh siswa. Penelitian ini berfokus pada menghasilkan soal secara otomatis dari sebuah kalimat.	NLP, k-nearest neighbor (kNN) classifier	WordNet, Penn Treebank II, Conditional Random Field (CRF), Google AJAX Search API, PHP, AJAX	Dengan diuji oleh <i>expert</i> untuk menjawab soal hasil <i>generate</i> dari sistem, teks yang diberikan ada dua teks yaitu teks A teks B, dan teks C. Hasilnya adalah pengecoh untuk teks A sebesar 66,7% dan untuk teks B 85,7% terbukti secara tata bahasa tidak benar. Sedangkan untuk teks C lebih dari 60% pengecohnya adalah tidak bagus atau tidak pantas oleh lebih dari satu <i>expert</i> .	Multiple Choice Gap Fill
Yuan, et al (2017)	Peneliti mengusulkan <i>neural model</i> untuk <i>generates natural-language questions</i> dari dokumen. Menjelaskan bagaimana melakukan <i>train</i> untuk model menggunakan kombinasi dari <i>supervised</i> dan <i>reinforcement learning</i> .	<i>Supervised, reinforcement learning</i>	dataset SQuAD, neural machine translation (NMT), Seq2Seq System	Dengan membandingkan sistem yang peneliti kembangkan dengan sistem Seq2Seq, hasilnya adalah dilihat dari nilai Fluency (PPL), sistem peneliti memiliki nilai 175,7 dan Seq2Seq memiliki nilai 153,2. Sistem yang dikembangkan oleh peneliti menghasilkan soal yang lebih spesifik dibanding dengan Seq2Seq, akan tetapi tetap ada kekurangannya dalam konteksnya.	Short Answer Question

Sitasi	Tujuan	Metode	Tools	Hasil	Jenis Soal
Mazidi & Nielsen (2015)	Hasil pencapaian pembelajaran yang dilakukan oleh siswa dapat dilihat dari jawaban pertanyaannya, untuk mengukur seberapa apa pembelajaran yang siswa lakukan. Oleh karena hal itu peneliti mengembangkan sistem pembuat soal otomatis untuk mengukur pembelajaran siswa.	semantic pattern recognition	SENNA, NLTK, python, H&S, LPN&W	Hasil evaluasi dari sistem ini menunjukkan 44% pengurangan pada tingkat kesalahan terhadap sistem sebelumnya yaitu H&S, dan LPN&W. Naik sampai 61% pengurangan pada tingkat kesalahan di penilaian tata bahasa.	Short Answer Question
Liu, et al (2013)	Penelitian ini bermaksud dapat membuat sistem yang dapat <i>men-support</i> siswa dalam pembelajaran. Yang tujuan dari penghasil soal otomatis ini pada <i>reading comprehension</i> atau <i>vocabulary assessment</i> .	NLP, Latent Semantic Analysis (LSA)	Writers Workshop, SaK, Sourcer's Apprentice Intelligent Feedback system (SAIF), intelligent tutoring system (ITS)	Menghasilkan sistem penghasil soal otomatis yang diberi nama G-Ask. Peneliti melakukan evaluasi dengan menggunakan <i>data training</i> dari 45 paper, dan 33 literatur review paper untuk melakukan testing 534 sitasi dan 469 hasil ekstraksi. <i>Citation Extraction Rate</i> sebesar 88%	Short Answer Question

Sitasi	Tujuan	Metode	Tools	Hasil	Jenis Soal
Chali & Hasan (2012)	Membuat penghasil soal otomatis dari teks yang memiliki topik. Dengan memakai Latent Dirichlet Allocation (LDA) untuk mengidentifikasi sub-topik.	Latent Dirichlet Allocation (LDA), Natural Language Processing (NLP), Natural Language Generation (NLG), Named Entity (NE), Semantic Role Labeling (SRL)	extended string subsequence kernel (ESSK), ASSERT, Word Sequence Kernel (WSK) , String Subsequence Kernel (SSK)	Ada baseline 1 dan baseline 2 yang dengan maksud membuat QG sistem dengan dua cara. Hasilnya adalah pada baseline 1 skor untuk relevansi dari topik sebesar 2,15 dan sintaknya 2,63. Sedangkan untuk baseline 2, skor relevansinya sebesar 3,24 dan sintaknya 3,30. QG sistem yang dibuat oleh peneliti menghasilkan skor relevansi sebesar 3,48 dan sintaknya 3,55.	Short Answer Question
Ali, et al (2010)	Pada penelitian ini membuat sistem penghasil soal otomatis dari kalimat. Yang akan <i>generate</i> pertanyaan yang mungkin dapat digunakan. Dengan modul atau proses dalam pemrosesan data.	Syntactic Parsing, POS Tagging, Named Entity Tagging	WTML (Web Testing Markup Language), HTML, Question Generation Shared Task Evaluation Challenge 2010 (QGSTEC), Elementary Sentence Extraction System (ESES)	Evaluasi sistem yang dikembangkan memakai QGSTEC. Tipe soal yang di <i>generate</i> berupa 5W+1H. Hasilnya adalah <i>recall</i> nya rata-rata dibawah 0,300.	Short Answer Question

Sitasi	Tujuan	Metode	Tools	Hasil	Jenis Soal
Kim & Kim (2008)	Karena kesulitan dalam pembuatan sistem <i>question-answering</i> (QA) adalah di <i>generate natural language</i> . Maka penelitian ini bertujuan untuk memberikan <i>style</i> baru dalam arsitektur QA, dengan menggunakan kalimat pada dokumen sebagai sumber dari QA.	NLP, Named Entity Recognition (NER)	Phyton, MySQL DBMS, NER tool	Hasil dari penelitian ini adalah sistem QA yang dapat membuat pertanyaan dari paragraf. Dengan metode penelitiannya adalah <i>sentence split</i> , NER, <i>question generation</i> , <i>question filtering</i> , dan <i>question/answer indexing</i> .	Short Answer Question

Pada Tabel 4.19 menunjukkan beberapa penelitian mengenai *automatic question generation* yang sudah dilakukan. Penelitian tersebut yang akan dibandingkan dengan penelitian yang dilakukan oleh peneliti. Adapun untuk penjelasan mengenai perbandingan adalah sebagai berikut.

1. Jenis Soal

Jenis soal yang dihasilkan dalam penelitian ini yaitu *short answer question* yang ada pada bagian *reading comprehension* IELTS. Seperti pada penelitian yang dilakukan oleh Yuan, dkk (2017), menghasilkan soal dengan jenis *short answer question*, akan tetapi metode yang dilakukan berbeda dengan penelitian ini, metode yang dilakukan yaitu menggunakan *supervised reinforcement learning* sedangkan dalam penelitian menggunakan metode *natural language processing*. Begitupun soal yang dihasilkan tidak dikhususkan untuk IELTS, akan tetapi soal yang dihasilkan berasal dari paragraf, sama seperti yang dilakukan oleh penelitian ini.

Begitu juga seperti penelitian yang dilakukan oleh Mazidi dan Nielsen (2015), untuk jenis soal yang dihasilkan sama akan tetapi soal yang dihasilkan tidak dikhususkan untuk soal IELTS. Metode yang dipakai pun berbeda, yaitu *semantic pattern recognition* dengan memakai *tools* yaitu SENNA, NLTK, H&S, dan LPN&W. H&S dan LNP&W digunakan dalam penelitian mereka untuk melakukan evaluasi atau perbandingan antara soal yang dihasilkan dengan sistem H&S dan juga LPN&W. Sedangkan dalam penelitian yang dilakukan oleh Liu, dkk (2013), untuk jenis soal yang dihasilkan sama dengan penelitian ini, tujuan dari soal yang dihasilkan sama yaitu untuk *reading comprehension* akan tetapi seperti penelitian sebelumnya, tidak dikhususkan untuk IELTS. Metode yang digunakan yaitu *natural language processing* dan juga *latent semantic analysis*.

Penelitian yang dilakukan oleh Mazidi dan Tarau (2016), jenis soal yang dihasilkan berbeda, yaitu soal yang dihasilkan berupa *multiple choice* yang kemudian dilakukan evaluasi yaitu membandingkan soal yang dihasilkan dengan sistem yang sudah ada pada penelitian sebelumnya yaitu sistem H&S. Selain itu, penelitian yang dilakukan oleh Kumar, dkk (2015) juga menghasilkan jenis soal yang berbeda yaitu *gap fill* dengan menggunakan metode *natural language processing*, *human intelligence tasks*, dan juga algoritma *support vector machine*.

2. Metode

Metode yang digunakan dalam penelitian-penelitian sebelumnya berbeda-beda untuk membuat *automatic question generation*. Seperti penelitian yang dilakukan oleh Majumder dan Saha (2014), dalam penelitiannya metode yang digunakan yaitu antara lain *Support Vector Machine (SVM) classifier*, *Parse Tree Matching (PTM) Algorithm*, *Conditional Random Field (CRF)*, jenis soal yang dihasilkan dalam penelitian tersebut adalah *multiple choice* dan soal yang dihasilkan tidak dikhususkan untuk bagian *reading comprehension* IELTS. Dalam penelitian yang dilakukan oleh Alsubait, dkk (2012), jenis soal yang dihasilkan sama seperti penelitian yang dilakukan oleh Majumder dan Saha (2014), akan tetapi metode dari penelitian yang dilakukan yaitu *Structure Mapping Theory (SMT)*, dan *Vector Space Model (VSM)*.

Sedangkan dalam penelitian yang dilakukan oleh Kim dan Kim (2008), jenis soal yang dihasilkan sama dengan penelitian ini yaitu soal dengan jenis *short answer question*. Metode yang digunakan pun mirip, yaitu menggunakan metode *natural language processing* dan juga *name entity recognition*, akan tetapi *tools* yang dipakai berbeda. Beda lagi dengan penelitian yang dilakukan oleh Chali dan Hasan (2012), metode yang digunakan berbeda walaupun jenis soal yang dihasilkan sama seperti penelitian ini yaitu jenis soal *short answer question*. Adapun metode yang digunakan yaitu antara lain *Latent Dirichlet Allocation (LDA)*, *Natural Language Processing (NLP)*, *Natural Language Generation (NLG)*, *Named Entity (NE)*, *Semantic Role Labeling (SRL)*. LDA ini digunakan untuk melakukan identifikasi sub-topik yang kemudian menghasilkan soal dari topik tersebut.

3. Tools

Dalam pemakaian *tools* yang digunakan untuk mengembangkan sistem *automatic question generation* ada berbagai macam, seperti penelitian yang dilakukan oleh Goto, dkk (2010) untuk *tools* yang digunakan antara lain WordNet, *Penn Treebank II*, *Conditional Random Field (CRF)*, *Google AJAX Search API*, dan PHP. Pemakaian WordNet ini sama dengan penelitian ini, yaitu digunakan untuk mendapatkan dataset kata atau jika dalam penelitian ini digunakan pada saat penentuan sinonim, yaitu mencari sinonim dari suatu kata, akan tetapi bahasa pemrograman yang digunakan berbeda. Beda lagi dengan penelitian yang dilakukan

oleh Huang, dkk (2012), *tools* yang digunakan dalam pengembangan sistem *automatic question generation* ini yaitu *Tgrep2 patterns* dengan sumber teks antara lain *Time For Kids*, *Student Times*, *Voice of America*, *CNN*, *China Post Online* dan *Yahoo! News*. Adapun kedua penelitian tersebut menghasilkan soal dengan jenis *gap fill* dengan pilihan ganda.

Berbeda dengan dua penelitian tersebut, penelitian yang dilakukan oleh Kumar, dkk (2015), menghasilkan soal dengan jenis *gap fill*, menggunakan *tools* antara lain *Amazon Mechanical Turk*, *Word2Vec*, *WordNet*, *TF-IDF search engine*, dan bahasa pemrograman *python*. Penelitian tersebut menghasilkan sebuah sistem *automatic question generation* yang diberi nama *RevUp*. Pemakaian *WordNet* dan juga *Word2Vec* ini sama dengan penelitian ini, akan tetapi dalam penelitian ini digunakan dalam proses pengubahan sinonim, yang mana *WordNet* digunakan untuk mendapatkan sinonim dari kata dan *Word2Vec* digunakan untuk menentukan jarak antara kata asli dengan sinonim, sehingga dapat didapatkan kata sinonim yang paling mendekati atau mirip dengan kata aslinya.

4. Hasil

Hasil dari penelitian mengenai *automatic question generation* pun berbeda-beda, cara untuk melakukan evaluasi pun berbeda-beda, disesuaikan dengan jenis soal yang dihasilkan dan juga tujuan dari penelitian dilakukan. Seperti penelitian yang dilakukan oleh Heilman dan Eskenazi (2007), soal yang dihasilkan oleh sistem memiliki presisi sebesar 80%, penelitian ini menggunakan metode *thesaurus extraction technique*, evaluasi yang digunakan pun menggunakan *expert* dalam menilai soal yang dihasilkan oleh sistem. Penelitian ini menghasilkan soal dengan jenis pilihan ganda.

Berbeda dengan penelitian yang dilakukan oleh Heilman dan Eskenazi (2007), penelitian yang dilakukan oleh Yuan, dkk (2017) menghasilkan soal dengan jenis yang sama dengan penelitian ini, yaitu jenis soal *short answer question*. Akan tetapi evaluasi yang dilakukan berbeda, pada penelitiannya evaluasi dilakukan dengan cara membandingkan sistem yang dikembangkan dengan sistem yang sudah ada pada penelitian sebelumnya yaitu *Seq2Seq* dan hasilnya memakai nilai *Fluency* (PPL). Nilai yang dihasilkan oleh peneliti yaitu sebesar 175,7 sedangkan sistem *Seq2Seq* memiliki nilai sebesar 153,2. Yang artinya sistem yang dikembangkan

oleh peneliti menghasilkan soal yang lebih spesifik, walaupun tetap ada kekurangan dalam hal konteks.